Generalized k-means based clustering for temporal data under time warp

Saeid Soheily-Khah

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique et Mathématiques appliquées**

Arrêté ministériel : 7 août 2006

Présentée par
Saeid SOHEILY-KHAH

Thèse dirigée par **Ahlame DOUZAL** et
codirigée par **Eric GAUSSIER**

préparée au sein du
Laboratoire d'Informatique de Grenoble (LIG)
dans **l'école doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique (MSTII)**

Generalized k-means based clustering for temporal data under time warp

Thèse soutenue publiquement le **7 OCT 2016**,
devant le jury composé de:

Mohamed NADIF
Professeur, LIPADE, University of Paris Descartes, Rapporteur
Paul HONEINE
Professeur, LITIS, Université de Rouen Normandie, Rapporteur
Pierre-François MARTEAU
Professeur, IRISA, Université Bretagne Sud, Examinateur, Président du jury
Jose Antonio VILAR FERNANDEZ
Maître de conférences, MODES, Universidade da Coruña, Examinateur
Ahlame DOUZAL
Maître de conférences (HDR), LIG, Université Grenoble Alpes, Directeur de thèse
Eric GAUSSIER
Professeur, LIG, Université Grenoble Alpes, Co-Directeur de thèse

UNIVERSITÉ DE GRENOBLE

ÉCOLE DOCTORALE MSTII

Description de complète de l'école doctorale

T H È S E

pour obtenir le titre de

docteur en sciences

de l'Université de Grenoble-Alpes

Mention : Informatique et Mathématiques appliquées

Présentée et soutenue par

Saeid SOHEILY-KHAH

Generalized k-means based Clustering for Temporal Data under Time Warp

Thèse dirigée par Ahlame DOUZAL et
codirigée par Eric GAUSSIER

préparée au Laboratoire d'Informatique de Grenoble (LIG)

soutenue le 7 OCT 2016

Jury :

Rapporteurs :	Mohamed NADIF	- LIPADE, University of Paris Descartes
	Paul HONEINE	- LITIS, Université de Rouen Normandie
Directeur :	Ahlame DOUZAL	- LIG, Université Grenoble Alpes
Co-Directeur :	Eric GAUSSIER	- LIG, Université Grenoble Alpes
Président :	Pierre-François MARTEAU	- IRISA, Université Bretagne Sud
Examinateur :	Jose Antonio VILAR FERNANDEZ	- MODES, Universidade da Coruña
	Pierre-François MARTEAU	- IRISA, Université Bretagne Sud

Acknowledgments

First and foremost, I would like to express my gratitude to my dear adviser, Ahlame DOUZAL, for her support, patience, and encouragement throughout my graduate studies. It has been an honor to be her Ph.D. student. It is not often that one finds an adviser and colleague that always finds the time for listening to the little problems and roadblocks that unavoidably crop up in the course of performing research. Her technical and editorial advice was essential to the completion of this dissertation and has taught me innumerable lessons and insights on the workings of academic research in general. She has been great teacher who has prepared me to get to this place in my academic life.

I have been extremely lucky to have a co-supervisor, Eric GAUSSIER, who cared about my work, and who responded to my questions and queries so promptly. I appreciate his wide knowledge and his assistance in writing reports (i.e., conference papers, journals and this thesis). He provided me with direction, technical support and became more of a mentor and friend, than a professor.

I would like to thank my dissertation committee members Mohamed NADIF, Paul HONEINE, Pierre-François MARTEAU and Jose Antonio VILAR FERNANDEZ for their interest in my work. They have each provided helpful feedback.

Completing this work would have been all the more difficult were it not for the support and friendship provided by the other members of the AMA team in Laboratoire d'Informatique de Grenoble. I am indebted to them for their help.

I want to thank my parents, who brought a 80-86 computer into the house when I was still impressionable enough to get sucked into a life of computer science. My parents, Ali and Safa, receive my deepest gratitude and love for their dedication and the many years of support during my undergraduate studies that provided the foundation for this work. I doubt that I will ever be able to convey my appreciation fully, but I owe them my eternal gratitude.

Last, but not least, I would like to thank my wife, Zohreh, for standing beside me throughout my career and writing this thesis, and also for her understanding and love during the past few years. She has been my inspiration and motivation for continuing to improve my knowledge and move my career forward. Her support and encouragement was in the end what made this dissertation possible. She is my rock, and I dedicate this thesis to her.

Contents

List of Figures

List of Tables

Table of Acronyms

LIG	*Laboratoire d'Informatique de Grenoble*
AMA	*Apprentissage, Méthode et Algorithme*
DTW	*Dynamic Time Warping*
NLAAF	*NonLinear Alignment and Averaging Filters*
PSA	*Prioritized Shape Averaging*
CWRT	*Cross-Words Reference Template*
DBA	*Dtw Barycebter Averaging*
SDTW	*Scaled Dynamic Time Warping*
COR	*Pearson CORrelation coefficient*
CORT	*Temporal CORrelation coefficient*
DACO	*Difference between Auto-Correlation Operators*
PAA	*Piecewise Aggregate Approximation*
PDTW	*Piecewise Dynamic Time Warping*
DTW$_{sc}$	*Sakoe-Chiba Dynamic Time Warping*
DTAK	*Dynamic Time Alignment Kernel*
PAM	*Partitioning Around Medoids*
CLARA	*Clustering LARge Applications*
MDS	*Multi Dimensional Scaling*
MCA	*Multiple Correspondence Analysis*
CA	*Correspondence Analysis*
RBF	*Radial Basis Function*
K$_{CORT}$	*Temporal Correlation Coefficient Kernel*
K$_{DACO}$	*Auto-correlation Kernel*
K$_{DTW}$	*Dynamic Time Warping Kernel*
K$_{SC}$	*Sakoe-Chiba Dynamic Time Warping Kernel*

K_{GDTW}	*Gaussian Dynamic Time Warping Kernel*
$\mathsf{K}_{GDTW_{SC}}$	*Sakoe-Chiba Gaussian Dynamic Time Warping Kernel*
K_{DTAK}	*Dynamic Temporal Alignment Kernel*
$\mathsf{K}_{DTAK_{SC}}$	*Sakoe-Chiba Dynamic Temporal Alignment Kernel*
K_{GA}	*Global Alignment Kernel*
K_{TGA}	*Triangular Global Alignment Kernel*
WDTW	*Weighted Dynamic Time Warping*
WK_{GDTW}	*Weighted Gaussian Dynamic Time Warping Kernel*
WK_{DTAK}	*Weighted Dynamic Temporal Alignment Kernel*
WK_{GA}	*Weighted Global Alignment Kernel*
WK_{TGA}	*Weighted Triangular Global Alignment Kernel*

Introduction

Due to rapid increase in data size, the idea of discovering hidden information in datasets has been exploded extensively in the last decade. This discovery has been centralized mainly on data mining, classification and clustering. One major problem that arises during the mining process is handling data with temporal feature. Temporal data naturally arise in various emerging applications as sensor networks, dynamic social networks, human mobility or internet of things. Temporal data refers to data, where changes over time or temporal aspects play a central role or are of interest. Unlike static data, there is high dependency among time series or sequences and the appropriate treatment of data dependency or correlation becomes critical in any temporal data processing.

Temporal data mining has recently attracted great attention in the data mining community [BF98]; [HBV01]; [KGP01]; [KLT03]. Basically temporal data mining is concerned with the analysis of temporal data and for finding temporal patterns and regularities in sets of temporal data. Since temporal data mining brings together techniques from different fields such as statistics, machine learning and databases, the literature is diffused among many different sources. According to the techniques of data mining and theory of statistical time series analysis, the theory of temporal data mining may involve the following areas of investigation since a general theory for this purpose is yet to be developed [LOW02]. Temporal data mining tasks include: characterization and representation, similarity computation and comparison, temporal association rules, clustering, classification, prediction and trend analysis, temporal pattern discovery, etc.

Clustering temporal data is a fundamental and important task, usually applied prior to any temporal data analysis or machine learning tasks, for summarization, extraction groups of time series and highlight the main underlying dynamics; all the more crucial for dimensionality reduction in big data context. Clustering problem is about partitioning a given dataset into groups or clusters such that the data points in a cluster are more similar to each other than data points in different clusters [GRS98]. It may be found under different names in different contexts, such as "unsupervised learning" in pattern recognition, "numerical taxonomy" in biology, "typology" in social sciences and "partition" in graph theory [TK99]. There are many applications where a temporal data clustering activity is relevant. For example in web activity logs, clusters can indicate navigation patterns of different user groups. In financial data, it would be of interest to group stocks that exhibit similar trends in price movements. Another example could be clustering of biological sequences like proteins or nucleic acids, so that sequences within a group have similar functional properties [F.88]; [Mil+99]; [Osa+02]. Hense, there are a variety of methods for clustering of temporal data and many of usual clustering methods need to compare the time series. Most of these comparison methods are derived from the Dynamic Time Warping (DTW) and align the observations of pairs of time series to identify the delays that can occur between times. Temporal clustering analysis provides an effective manner to discover the inherent structure and condense information over temporal data by exploring dynamic regularities underlying temporal data in an unsupervised

learning way. In particular, recent empirical studies in temporal data mining reveal that most of the existing clustering algorithms do not work well due to their complexity of underlying structure and data dependency [KK02], which poses a real challenge in clustering temporal data of a high dimensionality, complicated temporal correlation, and a substantial amount of noise.

Earlier, the clustering techniques have been extensively applied in various scientific areas. k-means-based clustering, viz. standard k-means [Mac67], k-means++ and all its variations, is among the most popular clustering algorithms, as it provides a good trade-off between the quality of obtained solution and its computational complexity [AV07]. However, time series k-means clustering, under the commonly used dynamic time warping (DTW) [SC78]; [KL83] or several well-established temporal kernels (e.g. κ_{DTAK}, κ_{GA}) [BHB02]; [Shi+02]; [Cut+07]; [Cut11], is challenging as estimating cluster centroids requires aligning multiple time series simultaneously. Under temporal proximity measures, one standard way to estimate the centroid of two time series is to embed the series into a new Euclidean space according to the optimal mapping between them. To average more than two time series, the problem becomes more complex as one needs to determine a multiple alignment that link simultaneously all the time series. That multiple alignment defines, similarly, a new Euclidean embedding space where time series can be projected and the global centroid estimated. Each dimension identifies, this time, a hyper link that connects more than two elements. Such alignments, referred to as multiple sequence alignment, become computationally prohibitive and impractical when the number of time series and their length increase [THG94]; [WJ94]; [CNH00].

To bypass the centroid estimation problem, costly k-medoids [KR87] and kernel k-means [Gir02]; [DGK04] are generally used for time series clustering [Lia05]. For k-medoids, a medoid (*i.e.* the element that minimizes the distance to the other elements of the cluster), is a good representative of the cluster mainly when time series have similar global dynamics within the class; it is however a bad representative for time series that share only local features [FDCG13]. For kernel k-means [Gir02]; [DGK04], as centroids can not be estimated in the Hilbert space, pairwise comparisons are used to assign a given time series to a cluster. While k-means, of linear complexity, remains a fast algorithm, k-medoids and kernel k-means have a quadratic complexity due to the pairwise comparisons involved.

This work proposes a fast, efficient and accurate approach that generalizes the k-means based clustering algorithm for temporal data based on i) an extension of the standard time warp measures to consider both global and local temporal differences and ii) a tractable and fast estimation of the cluster representatives based on the extended time warp measures. Temporal data centroid estimation is formalized as a non-convex quadratic constrained optimization problem, while all the popular existing approaches are of heuristic nature, with no guarantee of optimality. The developed solutions allow for estimating not only the temporal data centroid but also its weighting vector, which indicates the representativeness of the centroid elements. The solutions are particularly studied under the standard Dynamic Time Warping (DTW) [KL83], Dynamic Temporal Alignment Kernel (κ_{DTAK}) [Shi+02] and Global Alignment kernels (κ_{GA} and κ_{TGA}) [Cut+07]; [Cut11]. A wide range of public and challenging datasets, which are non-isotropic (*i.e.*, non-spherical), not well-isolated (and

thus non linearly separable), are used to compare the proposed generalized k-means with k-medoids and kernel k-means. The results of this comparison illustrate the benefits of the proposed method, which outperforms the alternative ones on all datasets. The impact of isotropy and isolation of clusters on the effectiveness of the clustering methods is also discussed.

The main contributions of the thesis are:

- Extend commonly used time warp measures: dynamic time warping, which is a dissimilarity measure, dynamic time warping kernel, temporal alignment kernel, and global alignment kernel, which are three similarity measures, to capture both global and local temporal differences.
- Formalize the temporal centroid estimation issue as an optimization problem and propose a fast and tractable solution under extended time warp measures.
- Propose a generalization of the k-means based clustering for temporal data under the extended time warp measures.
- Show through a deep analysis on a wide range of non-isotropic, linearly non-separable public and challenging datasets that the proposed solutions are faster and outperforms alternative methods, through (a) their efficiency on clustering and (b) the relevance of the centroids estimated.

In the remainder of the thesis, we use bold, lower-case letters for vectors, time series and alignments, the context being clear to differentiate between these elements.

Organisation du manuscrit

This manuscript is organized in four chapters as follows. The first chapter presents the major time series metrics, notations and definitions. As the most conventional metrics between time series are based on the concept of alignment, we then present the definition of temporal alignment, prior to introduce kernels for times series. In the second chapter, we will discuss about time series averaging and state-of-the-art of centroid estimation approaches. To do so, we present the consensus sequence and multiple alignment problem for time series averaging under time warp and review the related progressive and iterative approaches for centroid estimation. We show the importance of proposing a tractable and fast centroid estimation method. In the next chapter, we first study the k-means based clustering algorithms and its variations, their efficiency and complexities. While k-means based clustering is among the most popular clustering algorithms, it is a challenging task because of centroid estimation that needs to deal with the tricky multiple alignment problem under time warp. We then proposes the generalized k-means based clustering for time series under time warp measures. For this, we introduce motivation of this generalization, prior to the problem formalization. We give then an extension of the standard time warp measures and discuss about their properties. In the following, we describe the fully formalized centroid estimation procedure based on the extended time warp measures. We present the generalized centroid-based clustering for temporal data in the context of k-means under both dissimilarity and similarity measures, but as shown

later, the proposed solutions being directly applicable to any other centroid-based clustering algorithms. Lastly, the conducted experimentation and results obtained are discussed in the chapter four and we show through a deep analysis on a wide range of non-isotropic, linearly non-separable public data that the proposed solutions outperforms the alternative methods. The quantitative evaluation is finally completed by the qualitative comparison of the obtained centroids to the one of the time series they represent (i.e., ground truth). In the conclusion, we summarize the contributions of this thesis and discuss possible perspectives obtained from the thesis.

Notations

X	a set of data
$\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$	a set of time series
N	number of time series in a set
T	time series length
$\mathbb{S}$	space of all time series
$\mathbf{x}$	a time series
t	a time stamp
$\mathbf{s}$	similarity index
$\mathbf{d}$	dissimilarity index
$\mathcal{D}$	distance index
d_E	Euclidean distance
L_p	Minkovski p-norm
$\|\|\mathbf{x}\|\|_p$	p-norm of the vector $\mathbf{x}$
$\boldsymbol{\pi}$	An alignment between two time series
$\mathcal{B}(.,.)$	a behavior-based distance between two time series
$\mathcal{V}(.,.)$	a value-based distance between two time series
$\mathcal{BV}(.,.)$	a behavior-value-based distance between two time series
$d(.,.)$	a distance function
$s(.,.)$	a similarity function
$\kappa(.,.)$	a kernel similarity function
K	gram matrix (kernel matrix)
k	number of clusters
$\mathbf{c}$	a centroid (an average sequence of time series)
$\mathbf{C}_i$	representative of a cluster
p	iteration numbers
$\Phi(.)$	a non-linear transformation
O	time complexity

Chapter 1

Comparison of time series: state-of-the-art

Sommaire

The temporal sequences and time series are two frequently discussed topics in the literature. The time series can be seen as a special case of sequences, where the order criteria is time. In this chapter, we initially review the state-of-the-art of comparison measurements between pairs of time series. As the most conventional metrics between time series are based on the concept of alignment, we present the definition of temporal alignment, prior to introducing the kernels for times series.

1.1 Introduction

What is a time series? A time series is a kind of sequence with an ordered set of observations where the order criteria is time. A large variety of real world applications, such as meteorology, marketing, geophysics and astrophysics, collect observations that can be represented as time series. The most obvious example for a time series is probably the stock prices over successive

trading days or hourly household electric power consumption over successive hours. But not only the industry and financial sector produce a large amount of such data. Social media platforms and messaging services record up to a billion daily interactions [Pir09], which can be treated as time series. Besides the high dimensions of these data, the medical and biological sector provide a great variety of time series, as gene expression data, electrocardiograms, growth development charts and many more. In general, an increasingly large part of worlds data is in the form of time series [MR05]. Although statisticians have worked with time series for a century, the increasing use of temporal data and its special nature have attracted the interest of many researchers in the field of data mining and analysis [MR05]; [Fu11].

Time series analysis includes methods for analyzing temporal data in order to discover the inherent structure and condense information over temporal data, as well as the meaningful statistics and other characteristics of the data. In the context of statistics, finance, geophysics and meteorology the primary goal of time series analysis is forecasting. In the context of signal processing, control engineering and communication engineering it is used for signal detection and estimation, while in the context of data mining, pattern recognition and machine learning time series analysis can be used for clustering, classification, anomaly detection as well as forecasting. One major problem that arises during time series analysis is comparison of time series or sequences. For this comparison, the most studied approaches rely on proximity measures and metrics. Thus, a crucial question is establishing what we mean by "(dis)similar" data objects (i.e. determining a suitable (dis)similarity measure between two objects). In the specific context of time series data, the concept of (dis)similarity measure is particularly complex due to the dynamic character of the series. Hence, different approaches to define the (dis)similarity between time series have been proposed in the literature and a short overview of well-established ones is presented below.

We start this section by defining the general notion of proximity measures. Next, we discuss major basic proximity measures of time series. Later, we introduce definition of different kernels proximity measures for time series, as well as the explanation of temporal alignment, where the most conventional proximity measures between time series and sequences are based on the concept of alignment.

1.2 Comparison of time series

Mining and comparing time series address a large range of challenges, among them: the meaningfulness of the distance, similarity and dissimilarity measures. They are widely used in many research areas and applications. Distances or (dis)similarity measures are essential to solve many signal processing, image processing and pattern recognition problems such as classification, clustering, and retrieval problems. Various distances, similarity and dissimilarity measures that are applicable to compare time series, are presented over the past decade. We start here by defining the general notion of proximity measures.

Similarity measure

Let X be a set of data. A function $\mathbf{S}$: $X \times X \to \mathbb{R}$ is called a ***similarity*** on X, it obeys to the following properties, $\forall\ x, y \in X$:

- non-negativity: $\mathbf{S}(x, y) \geq 0$
- symmetry: $\mathbf{S}(x, y) = \mathbf{S}(y, x)$
- if $x \neq y$, $\mathbf{S}(x, x) = \mathbf{S}(y, y) > \mathbf{S}(x, y)$

A similarity measure takes on large values for similar objects and zero for very dissimilar objects (e.g. in the context of cluster analysis). In general, the similarity interval is $[0, 1]$, where 1 indicates the maximum of similarity measure.

Dissimilarity measure

A function $\mathbf{D}$: $X \times X \to \mathbb{R}$ is called a ***dissimilarity*** on X if, for all $x, y \in X$, it holds the three fundamentals following properties:

- non-negativity: $\mathbf{D}(x, y) \geq 0$
- symmetry: $\mathbf{D}(x, y) = \mathbf{D}(y, x)$
- reflexivity: $\mathbf{D}(x, x) = 0$

In some cases, the proximity measurement takes minimum values when the time series are similar together. For this purpose, we introduce a new measure of proximity opposite of the similarity index, called dissimilarity measure. The main transforms used to obtain a dissimilarity $\mathbf{D}$ from a similarity $\mathbf{S}$ are:

- $\mathbf{D}(x, y) = (\mathbf{S}(x, x) - \mathbf{S}(x, y))$
- $\mathbf{D}(x, y) = (\mathbf{S}(x, x) - \mathbf{S}(x, y))\ /\ \mathbf{S}(x, y)$
- $\mathbf{D}(x, y) = (\mathbf{S}(x, x) - \mathbf{S}(x, y))^{1/2}$

Distance metric

Distance measures play an important role for (dis)similarity problem, in data mining tasks. Concerning a distance measure, it is important to understand if it can be considered metric. A function $\mathcal{D}$: $X \times X \to \mathbb{R}$ is called a ***distance*** metric on X if, for all x, y and $z \in X$, it holds the properties:

- non-negativity: $\mathcal{D}(x, y) \geq 0$
- symmetry: $\mathcal{D}(x, y) = \mathcal{D}(x, y)$
- reflexivity: $\mathcal{D}(x, y) = 0$, if and only if $x = y$
- triangle inequality: $\mathcal{D}(x, y) \leq \mathcal{D}(x, z) + \mathcal{D}(z, y)$

It's easy to prove that the non-negativity property is included in the last three properties. If any of these is not obeyed, then the distance is non-metric. Although having the properties of a metric is desirable, a (dis)similarity measure can be quite effective without being a metric. In the following, to simplify, we use broadly the term ***metric*** to reference both.

1.3 Times series basic metrics

Time series analysis is an active research area with applications in a wide range of fields. One key component in temporal analysis is determining a proper (dis)similarity metric between two time series. This section is dedicated to briefly describe the major well-known measures in a nutshell, which have been grouped into two categories: value-based and behavior-based metrics.

1.3.1 Value-based metrics

A first way to compare time series data involves concept of values and distance metrics, where time series are compared according to their values. This subsection relies on two standard well-known division: (a) without delays (e.g. Minkowski distance) and (b) with delays (e.g. Dynamic Time Warping).

1.3.1.1 Without delays

Euclidean- (or Euclidian) The most used distance function in many applications, which is commonly accepted as the simplest distances between sequences. The Euclidean distance d_E (L_2 norm) between two time series $\mathbf{x}_i = (x_{i1}, ..., x_{iT})$ and $\mathbf{x}_j = (x_{j1}, ..., x_{jT})$ of length T, is defined as:

$$d_E(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{t=1}^{T} (x_{i_t} - x_{j_t})^2}$$

Minkowski Distance- The generalization of Euclidean Distance is Minkowski Distance, called L_p norm. It is defined as:

$$d_{L_p}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt[p]{\sum_{t=1}^{T} (x_{i_t} - x_{j_t})^p}$$

where p is called the Minkowski order. In fact, for Manhattan distance $p = 1$, for the Euclidean distance $p = 2$, while for the Maximum distance $p = \infty$. All L_p-norm distances do not consider the time warp. Unfortunately, they don't correspond to the common understanding of what a time series sequences is, and can not capture flexible similarities.

1.3.1.2 With delays

Dynamic time warping (DTW)- Searching the best alignment that matches two time series is an important task for many researcher. One of the eminent techniques to execute this task is Dynamic Time Warping (DTW) was introduced in [RD62]; [Ita75]; [KL83] with application in speech recognition. It finds the optimal alignment between two time series, and captures flexible similarities by aligning the elements inside both sequences. Intuitively, the time series are warped non-linearly in the time dimension to match each other. Simply, it is a generalization of Euclidean distance which allows a non-linear mapping of one time series to another one by minimizing the distance between both. DTW does not require that the two time series data have the same length, and it can handle local time shifting by duplicating (or re-sampling) the previous element of the time sequence.

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ be a set of time series $\mathbf{x}_i = (x_{i1}, ..., x_{iT})$ assumed of length T[1]. An alignment $\boldsymbol{\pi}$ of length $|\boldsymbol{\pi}| = m$ between two time series $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as the set of m ($T \leq m \leq 2T - 1$) couples of aligned elements of $\mathbf{x}_i$ to elements of $\mathbf{x}_j$:

$$\boldsymbol{\pi} = ((\pi_1(1), \pi_2(1)), (\pi_1(2), \pi_2(2)), ..., (\pi_1(m), \pi_2(m)))$$

where $\boldsymbol{\pi}$ defines a warping function that realizes a mapping from time axis of $\mathbf{x}_i$ onto time axis of $\mathbf{x}_j$, and the applications π_1 and π_2 defined from $\{1, ..., m\}$ to $\{1, .., T\}$ obey to the following boundary and monotonicity conditions:

$$\begin{aligned} 1 = \pi_1(1) \leq \pi_1(2) \leq ... \leq \pi_1(m) = T \\ 1 = \pi_2(1) \leq \pi_2(2) \leq ... \leq \pi_2(m) = T \end{aligned}$$

and $\forall\, l \in \{1, ..., m\}$,

$$\begin{aligned} \pi_1(l+1) \leq \pi_1(l) + 1 \text{ and } \pi_2(l+1) \leq \pi_2(l) \quad &+ \quad 1, \\ (\pi_1(l+1) - \pi_1(l)) \;+\; (\pi_2(l+1) - \pi_2(l)) \quad &\geq \quad 1 \end{aligned}$$

Intuitively, an alignment $\boldsymbol{\pi}$ defines a way to associate all elements of two time series. The alignments can be described by paths in the $T \times T$ grid as displayed in Figure 1.1, that crosses the elements of $\mathbf{x}_i$ and $\mathbf{x}_j$. For instance, the green path aligns the two time series as: $((x_{i1},x_{j1}),(x_{i2},x_{j2}),(x_{i2},x_{j3}),(x_{i3},x_{j4}),(x_{i4},x_{j4}),(x_{i5},x_{j5}),(x_{i6},x_{j6}),(x_{i7},x_{j7}))$. We will denote $\mathcal{A}$ as the set of all possible alignments between two time series.

[1] One can make this assumption as dynamic time warping can be applied equally on time series of same or different lengths.

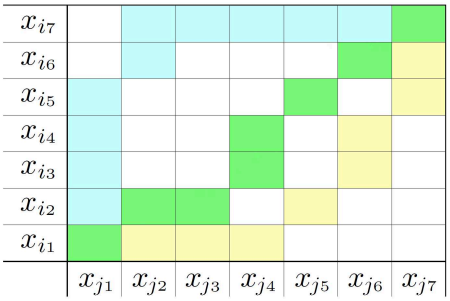


Figure 1.1: Three possible alignments (paths) between $\mathbf{x}_i$ and $\mathbf{x}_j$

Dynamic time warping is currently a well-known dissimilarity measure on time series and sequences, since it makes them possible to capture temporal distortions. The Dynamic Time Warping (DTW) between time series $\mathbf{x}_i$ and time series $\mathbf{x}_j$, with the aim of minimization of the mapping cost, is defined by:

$$\text{DTW}(\mathbf{x}_i, \mathbf{x}_j) = \min_{\boldsymbol{\pi} \in \mathcal{A}} \frac{1}{|\boldsymbol{\pi}|} \sum_{(t',t) \in \boldsymbol{\pi}} \varphi(x_{it'}, x_{jt}) \tag{1.1}$$

where $\varphi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ is a positive, real-valued, divergence function (generally Euclidean norm). Figure 1.2 shows the alignment between two sample time series with delay (dynamic time warping) in comparison with the alignment without delay (Euclidean).

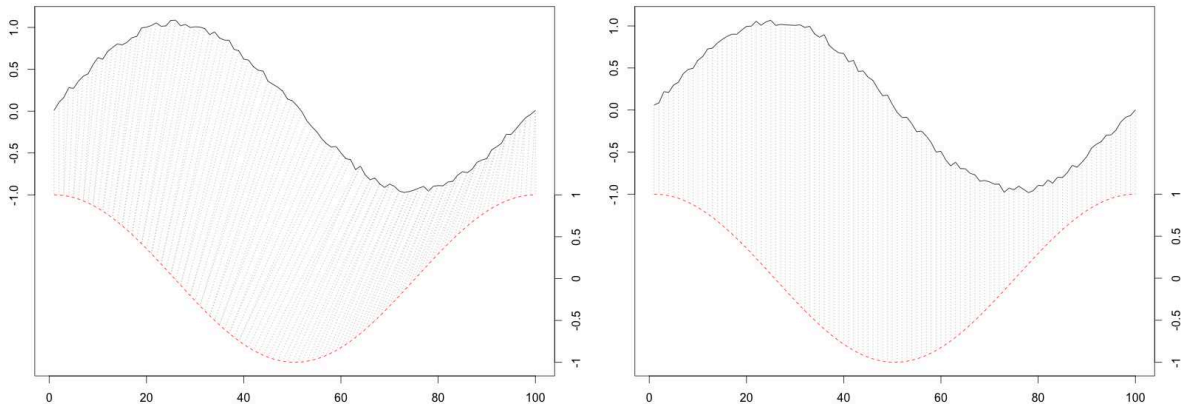


Figure 1.2: dynamic time warping alignment (left) vs. euclidean alignment (right)

While Dynamic Time Warping alignments deal with delays or time shifting (see Figure 1.2-left), the Euclidean alignment $\boldsymbol{\pi}$ between $\mathbf{x}_i$ and $\mathbf{x}_j$ aligns elements observed at the same time:

$$\boldsymbol{\pi} = ((\pi_1(1), \pi_2(1)), (\pi_1(2), \pi_2(2)), ..., (\pi_1(T), \pi_2(T)))$$

where $\forall t = 1, ..., T$, $\pi_1(t) = \pi_2(t) = t$, $|\boldsymbol{\pi}| = T$ (see Figure 1.2-right). According to the alignment definition, the euclidean distance (d_E) between the time series $\mathbf{x}_i$ and $\mathbf{x}_j$ is given

by:

$$d_E(\mathbf{x}_i, \mathbf{x}_j) \stackrel{\text{def}}{=} \frac{1}{|\boldsymbol{\pi}|} \sum_{k=1}^{|\boldsymbol{\pi}|} \varphi(x_{i_{\pi_1(k)}}, x_{j_{\pi_2(k)}}) = \frac{1}{T} \sum_{t=1}^{T} \varphi(x_{it}, x_{jt})$$

φ taken as the Euclidean norm.

Finally, Figure 1.3 shows the optimal alignment path between two sample time series with and without considering time warp. To recall, boundary, monotonicity and continuity are three important properties which are applied to the construction of the path. The boundary condition imposes that the first elements of the two time series are aligned to each other, as well as the last sequences elements. In other word, the alignment refers to the entire both time series sequences. Monotonicity preserve the time-ordering of elements. It means, the alignment path doesn't go back in "*time*" index. Continuity limits the warping path from long jumps and it guarantees that alignment does not omit important features.

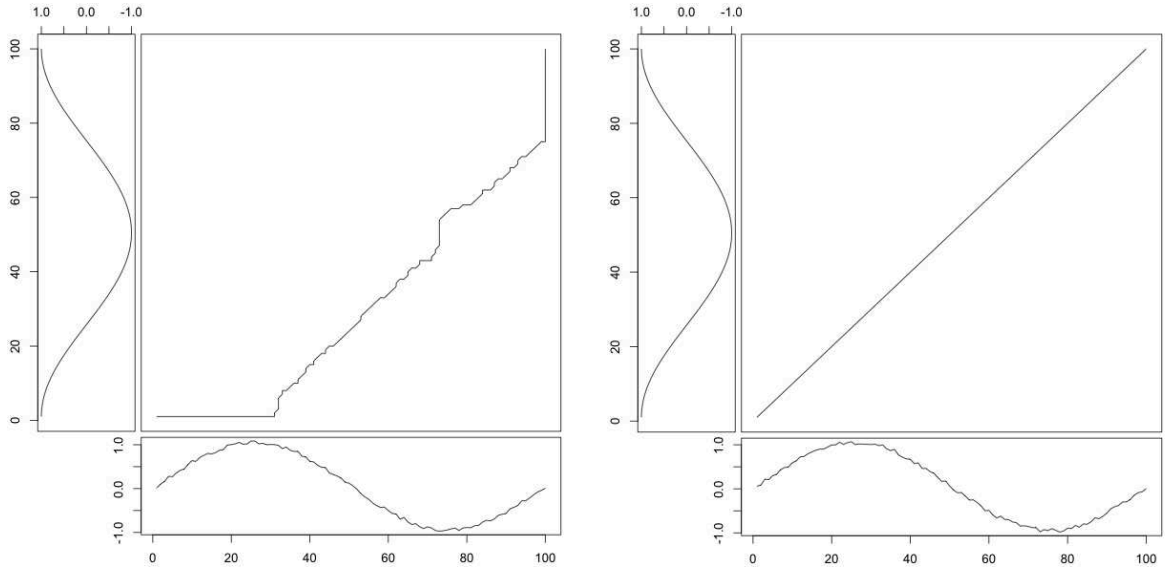


Figure 1.3: The optimal alignment path between two sample time series with time warp (left), without time warp (right)

A dynamic programming approach is used to find the minimum distance for alignment path. Let $\mathbf{x}_i$ and $\mathbf{x}_j$ two time series. A two-dimensional $|\mathbf{x}_i|$ by $|\mathbf{x}_j|$ cost matrix $\mathbf{D}$ is built up, where the value at $\mathbf{D}(t, t')$ is the minimun distance warp path that can be constructed from the two time series $\mathbf{x}_i = (x_{i1}, ..., x_{it})$ and $\mathbf{x}_j = (x_{j1}, ..., x_{jt'})$. Finally, the value at $\mathbf{D}(|\mathbf{x}_i|, |\mathbf{x}_j|)$ will contain the minimum distance between the times series $\mathbf{x}_i$ and time series $\mathbf{x}_j$ under time warp. Note that DTW is not a metric as not satisfy the triangle inequality.

Property 1.1
DTW does not follow triangle inequality.

As we can see, for instance, given three time series data, $\mathbf{x}_i = [0]$, $\mathbf{x}_j = [1, 2]$ and $\mathbf{x}_k = [2,3,3]$, then: $\text{DTW}(\mathbf{x}_i, \mathbf{x}_j) = 3$, $\text{DTW}(\mathbf{x}_j, \mathbf{x}_k) = 3$, $\text{DTW}(\mathbf{x}_i, \mathbf{x}_k) = 8$. Hense,

$$\text{DTW}(\mathbf{x}_i, \mathbf{x}_j) + \text{DTW}(\mathbf{x}_j, \mathbf{x}_k) \not\geq \text{DTW}(\mathbf{x}_i, \mathbf{x}_k). \square$$

Time and space complexity of the dynamic time warping is straightforward to define. Each cell in the cost matrix is filled once in constant time. The cost of the optimal alignment can be recursively computed by:

$$\mathbf{D}(t, t') = d(t, t') + \min \left\{ \begin{array}{l} \mathbf{D}(t-1, t') \\ \mathbf{D}(t-1, t'-1) \\ \mathbf{D}(t, t'-1) \end{array} \right\}$$

where d is a distance function between the elements of time series (e.g. Euclidean).

This yields complexity of $O(N^2)$. Dynamic time warping is able to find the optimal global alignment path between time series and is probably the most commonly used measure to assess the dissimilarity between them. Dynamic time warping (DTW) has enjoyed success in many areas where its time complexity is not an issue. However, conventional DTW is much too slow for searching an alignment path for large datasets. For this problem, the idea of the DTW technique for aligning time series is changing with the aim of improvement time and space complexity and accuracy. In general, the methods which make DTW faster divide in three main different categories: 1) constraints (e.g., Sakoe-Chiba band), 2) indexing (e.g., piecewise), and 3) data abstraction (e.g., multiscale). Here we briefly mentioned some major ideas that try to enhance this technique.

Constrained DTW

Constraints are widely used to speed up dynamic time warping algorithm. The most commonly used ones is the Sakoe-Chiba band (symmetric and asymmetric) [SC71]; [SC78] and Itakura parallelogram [Ita75], which are shown in Figure 1.4.

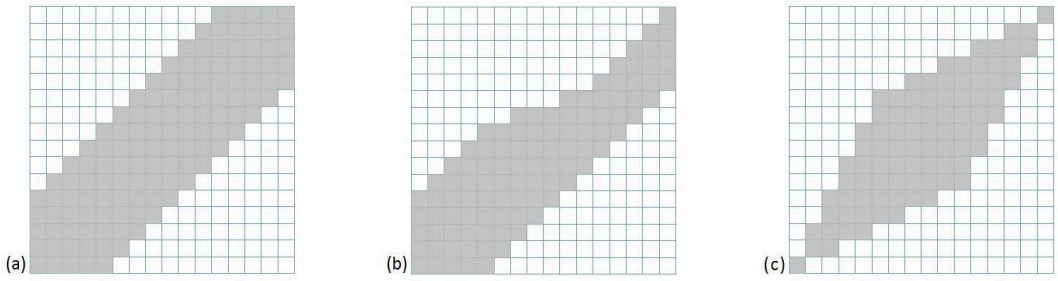


Figure 1.4: Speed up DTW using constraints: (a)Sakoe-Chiba band (b)Asymmetric Sakoe-Chiba band (c)Itakura parallelogram

The cost matrix are filled by the DTW algorithm in the shaded areas around the diagonal. In this case, the algorithm finds the optimal alignment warp math through the constraints

window. However, the global optimal alignment path will not be found, if it is not fully inside the window.

The Sakoe-Chiba dynamic time warping (DTW_{sc}) between two time series, which is the most well-used constrained DTW, is defined by:

$$\text{DTW}_{sc}(\mathbf{x}_i, \mathbf{x}_j) \stackrel{\text{def}}{=} \min_{\boldsymbol{\pi} \in \mathcal{A}} \mathbf{C}(\boldsymbol{\pi})$$

$$\mathbf{C}(\boldsymbol{\pi}) \stackrel{\text{def}}{=} \frac{1}{|\boldsymbol{\pi}|} \sum_{k=1}^{|\boldsymbol{\pi}|} w_{\pi_1(k),\pi_2(k)} \;\; \varphi(x_{i_{\pi_1(k)}}, x_{j_{\pi_2(k)}})$$

$$= \frac{1}{|\boldsymbol{\pi}|} \sum_{(t,t') \in \boldsymbol{\pi}} w_{t,t'} \;\; \varphi(x_{it}, x_{jt'})$$

where if $|t - t'| < c$ then $w_{t,t'} = 1$, and else ∞. Function φ taken as the euclidean norm and parameter c is the Sakoe-Chiba band width. Figure 1.5 shows examples of the cost matrix and the two constrained optimal alignment path between two sample time series.

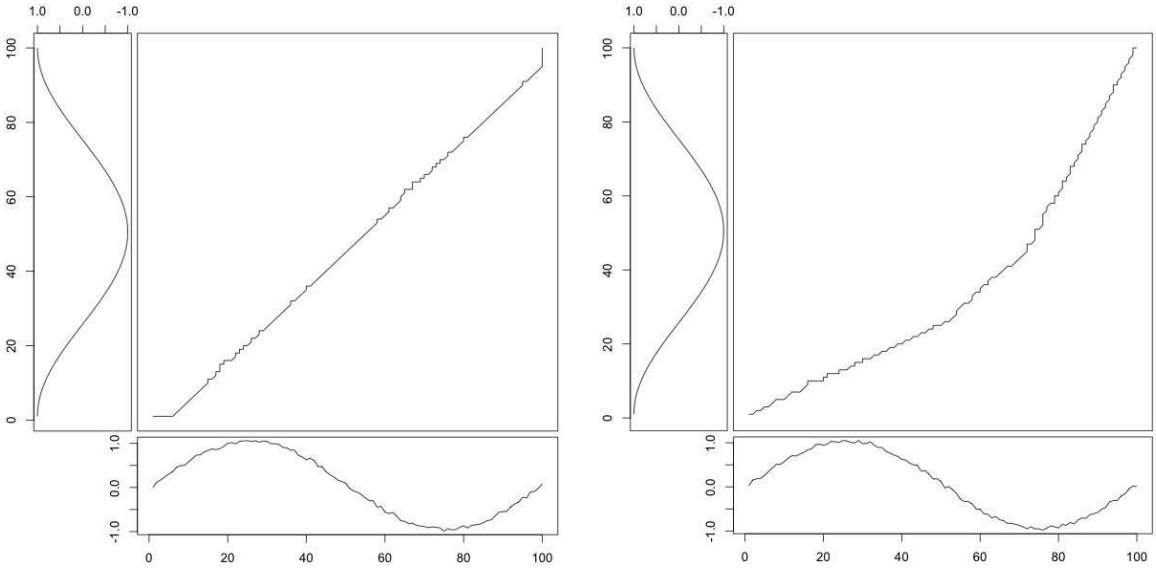


Figure 1.5: The constrained optimal alignment path between two sample time series (Sakoe-Chiba band = 5 - left), (Itakura parallelogram - right)

Piecewise DTW

Piecewise Dynamic Time Warping (PDTW) [KP00], takes advantage of the fact that one can efficiently approximate most of the time series by a Piecewise Aggregate Approximation (PAA). Let $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iT})$ be a time series sequence that we wish to reduce its dimensionality to T', where $(1 \leq T' \leq T)$. A time series $\mathbf{x}_i^*$ of length T' is represented by $\mathbf{x}_i^* = (x_{i1}, ..., x_{iT'})$, which t^{th} element of $\mathbf{x}_i^*$ is calculated by the following equation:

$$x^*_{it} = \frac{T'}{T} \sum_{t'=\frac{T}{T'}(t-1)+1}^{\frac{T}{T'}t} x_{it'}$$

Simply, to reduce the data from T dimensions to T' dimensions, the data is divided into T' frames with equal size. The average value of the data falling within each frame is computed and a vector of these values becomes the data reduced representation. The compression rate c ($c = \frac{T}{T'}$) is equal to the ratio of the length of the original time series to the length of its PAA representation. Choosing a value for c, is a trade-off between memory savings and accuracy [KP00]. Figure 1.6 illustrates a time series and its PAA approximation.

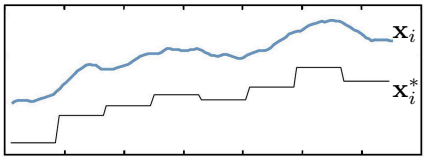


Figure 1.6: The time series sequence $\mathbf{x}_i$, and its PAA $\mathbf{x}^*_i$

Finally, Figure 1.7 illustrates strong visual evidence that PDTW finds alignment that are very similar to those produced by DTW. Hence, the time complexity for PDTW is $O(T'^2)$, which means the speedup obtained by PDTW should be $\frac{O(T^2)}{O(T'^2)}$ which is $O(c^2)$.

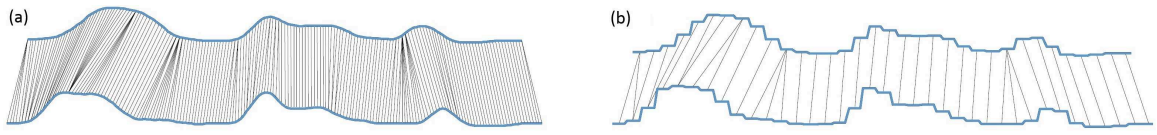


Figure 1.7: Alignment between two time series by (a)DTW (b)PDTW

Therefore, piecewise method by finding a time series which is most similar to a given one, significantly speed up many DTW applications by reducing the number of times which the DTW should be run.

MultiScale DTW (FastDTW)

To obtain an efficient as well as robust algorithm to compute DTW-based alignments, one can combine the global constraints and dimensionality reduction strategies in some iterative approaches to generate data-dependent constraint regions. The general strategy of multiscale DTW (or FastDTW)[SC04] is to recursively project a warping path computed at a coarse resolution level to the next level and then refine the projected warping path. Multiscale DTW or fastDTW speeds up the dynamic time warping algorithm by running DTW on a reduced representation of the data.

Let $\mathbf{x}_i = (x_{i1}, ..., x_{iT})$ and $\mathbf{x}_j = (x_{j1}, ..., x_{jT})$ be the sequences to be aligned, having lengths T. The aim is to compute an optimal warping path between two time series $\mathbf{x}_i$ and $\mathbf{x}_j$. The highest resolution level will be called as Level 1. By reducing the feature sampling rate by a factor of f, one achieves time series of length $\frac{T}{f}$. Next, one computes an optimal alignment warping path between two dimension-reduced time series on the resulting resolution level (Level 2). The obtained path is projected onto highest level (Level 1) and it defines a constraint region R. Lastly, an optimal alignment warping path relative to the restricted area R is computed. In general, the overall number of cells to be computed in this process is much smaller than the total number of cells on Level 1 (T^2). The constrained warping path may not go along with the optimal alignment warping path. To mitigate this problem, one can increase the constraint region R by adding δ cells to the left, right, top and bottom of every cell in R. The resulting region R^δ will be called as δ-neighborhood of R [ZM06]. Figure 1.8 demonstrates an example of multiscale DTW for alignment two time series in the direction of reduce time and space complexity.

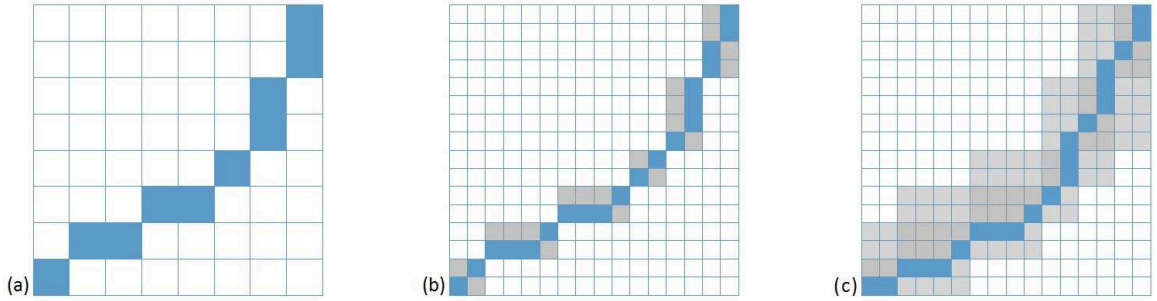


Figure 1.8: (a) The optimal alignment warp path on level 2. (b) The optimal alignment warp path with respect to the constraint region R by projecting alignment path to level 1. (c) The optimal alignment warp path with $\delta = 2$

Finally, in spite of the great success of DTW and its variants in a diversity of domains, there are still several persistent myths about it, as finding ways to speed up DTW with no (or relaxed) constraints.

1.3.2 Behavior-based metrics

The second category of time series metrics concerns behavior-based metrics, where time series are compared according to their behaviors regardless of their values. That is the case when time series of a same class exhibit similar shape or behavior, and time series of different classes have different shapes. Hence, comparing the time series on the base of their value may not be valid assumption. In this context, we should define which time series are more similar together and which ones are different. The definition of similar, opposite and different behaviors are given as following.

Similar Behavior. Time series $\mathbf{x}_i, \mathbf{x}_j$ are of **similar** behavior, if for each period $[t_i, t_{i+1}]$, $i \in \{1, ..., T\}$, they increase or decrease simultaneously with the same growth rate.

Opposite Behavior. Time series $\mathbf{x}_i, \mathbf{x}_j$ are of **opposite** behavior, if for each period $[t_i, t_{i+1}]$, $i \in \{1, ..., T\}$, when one time series increases, the other decreases with the same growth rate in absolute value and vice-versa.

Different Behavior. Time series $\mathbf{x}_i, \mathbf{x}_j$ are of **different** behavior, if they are not similar nor opposite (linearly and stochastically independent).

Main techniques to recover time series behaviors are: slopes and derivatives comparison, ranks comparison, Pearson and temporal correlation coefficient, and difference between auto-correlation operators. In the following, we briefly describe some well-used behavior-based metrics.

1.3.2.1 Pearson CORrelation coefficient (COR)

Let $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ be a set of time series $\mathbf{x}_i = (x_{i1}, ..., x_{iT})$, $i \in \{1, ..., N\}$. The correlation coefficient between sequences $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined by:

$$\mathrm{COR}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{t=1}^{T}(x_{i_t} - \overline{\mathbf{x}}_i)(x_{j_t} - \overline{\mathbf{x}}_j)}{\sqrt{\sum_{t=1}^{T}(x_{i_t} - \overline{\mathbf{x}}_i)^2}\ \sqrt{\sum_{t=1}^{T}(x_{j_t} - \overline{\mathbf{x}}_j)^2}} \tag{1.2}$$

and

$$\overline{\mathbf{x}}_i = \tfrac{1}{T}\sum_{t=1}^{T} x_{i_t}$$

Correlation coefficient was first introduced by Bravais and later shown by Pearson [Pea96] to be the best possible correlation between two time series. Till now, many applications in different domains such as speech recognition, system design control, functional MRI and gene expression analysis have used the Pearson correlation coefficient as a behavior proximity measure between time series sequences [Mac+10]; [ENBJ05]; [AT10]; [Cab+07]; [RBK08]. The Pearson correlation coefficient changes between -1 and $+1$. The case COR $= +1$, called perfect positive correlation, occurs when two time series sequences perfectly coincide, and the case COR$= -1$, called the perfect negative correlation, occurs when two time series behave completely opposite. COR$= 0$ shows that the time series sequences have different behavior. Note that the higher correlation doesn't conclude the similar dynamics.

1.3.2.2 Temporal CORrelation coefficient (CORT)

To cope with temporal data, a variant of Pearson correlation coefficient, considering temporal dependency within r, is proposed in [DCA12], called as Temporal CORrelation coefficient (CORT). The authors considered an equivalent formula for the correlation coefficient relying on pairwise values differences as:

$$\text{COR}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{t,t'}(x_{i_t} - x_{i_{t'}})(x_{j_t} - x_{j_{t'}})}{\sqrt{\sum_{t,t'}(x_{i_t} - x_{i_{t'}})^2}\sqrt{\sum_{t,t'}(x_{j_t} - x_{j_{t'}})^2}} \tag{1.3}$$

One can see that the Pearson correlation coefficient assumes the independence of data as based on the differences between all pairs of observations at $[t, t']$; unlike, the behavior proximity needs only to capture how they behave at $[t, t + r]$. Therefore, the correlation coefficient is biased by all of the remaining pairs of values observed at interval $[t, t']$ with $|t - t'| > r$, $r \in [1, ..., T - 1]$. Temporal correlation coefficient defined as:

$$\text{CORT}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{t,t'} m_{tt'}(x_{i_t} - x_{i_{t'}})(x_{j_t} - x_{j_{t'}})}{\sqrt{\sum_{t,t'} m_{tt'}(x_{i_t} - x_{i_{t'}})^2}\sqrt{\sum_{t,t'} m_{tt'}(x_{j_t} - x_{j_{t'}})^2}} \tag{1.4}$$

where $m_{tt'} = 1$ if $|t - t'| \leq r$, otherwise 0. CORT belongs to the interval $[-1, +1]$. The value $\text{CORT}(\mathbf{x}_i, \mathbf{x}_j) = 1$ means that in any observed period $[t, t']$, $\mathbf{x}_i$ and $\mathbf{x}_j$ have similar behaviors. The value $\text{CORT}(\mathbf{x}_i, \mathbf{x}_j) = -1$ means that in any observed period $[t, t']$, $\mathbf{x}_i$ and $\mathbf{x}_j$ have opposite behaviors. Lastly, $\text{CORT}(\mathbf{x}_i, \mathbf{x}_j) = 0$ indicates that the time series are stochastically linearly independent. Temporal correlation is sensitive to noise. So, the parameter r can be learned or fixed a priori, and for noisy time series higher value of r is advised.

Figure 1.9 illustrates the comparison of the Pearson correlation coefficient and the temporal correlation coefficient between three samples time series.

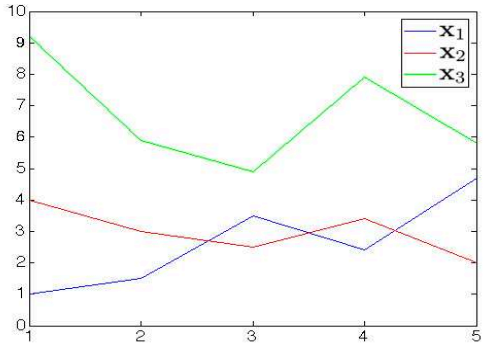


Figure 1.9: Example of comparison of COR vs CORT (r = 1). $\text{COR}(\mathbf{x}_1, \mathbf{x}_2) = -0.90$, $\text{COR}(\mathbf{x}_1, \mathbf{x}_3) = -0.65$, $\text{COR}(\mathbf{x}_2, \mathbf{x}_3) = 0.87$, $\text{CORT}(\mathbf{x}_1, \mathbf{x}_2) = -0.86$, $\text{CORT}(\mathbf{x}_1, \mathbf{x}_3) = -0.71$, $\text{CORT}(\mathbf{x}_2, \mathbf{x}_3) = 0.93$

1.3.2.3 Difference between Auto-Correlation Operators (DACO)

Auto-correlation is a representation of the degree of similarity which measures the dependency between a given time series and a shifted version of itself over successive time intervals. Let $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ be a set of time series $\mathbf{x}_i = (x_{i1}, ..., x_{iT})$ $i \in \{1, ..., N\}$. The DACO between the two time series $\mathbf{x}_i$ and $\mathbf{x}_j$ defined as [GHS11]:

$$\text{DACO}(\mathbf{x}_i, \mathbf{x}_j) = \|\tilde{\mathbf{x}}_\mathbf{i} - \tilde{\mathbf{x}}_\mathbf{j}\|^2 \tag{1.5}$$

where,

$$\tilde{\mathbf{x}}_\mathbf{i} = (\rho_\mathbf{1}(\mathbf{x_i}), ..., \rho_\mathbf{k}(\mathbf{x_i}))$$

and

$$\rho_\tau(\mathbf{x}_i) = \frac{\sum_{t=1}^{T-\tau}(x_{i_t} - \overline{\mathbf{x}}_i)(x_{i_{(t+\tau)}} - \overline{\mathbf{x}}_i)}{\sum_{t=1}^{T}(x_{i_t} - \overline{\mathbf{x}}_i)^2}$$

and τ is the time lag. Therefore, DACO compares time series by computing the distance between their dynamics, modeled by the auto-correlation operators. Note that the lower DACO doesn't represent the similar behavior.

1.3.3 Combined (Behavior-value-based) metrics

Considering the definition of time series basic metrics, the value-based one is based on the differences between the values of the time series and does not consider the dynamics and behaviors within the time series. Figure 1.10 shows two configurations: the left side, two time series $\mathbf{x}_1$ and $\mathbf{x}_2$ are definitely opposed, a decrease of one corresponding to a growth of the other and vice-versa. While the right side, the two series $\mathbf{x}_1$ and $\mathbf{x}_3$ are in the same direction, but have variations in intensity and scale. However, the overall behavior is similar. But, according to the definition of a value-based metrics, the time series $\mathbf{x}_1$ and $\mathbf{x}_2$ in left, have the same distance (in Euclidean) with the two time series $\mathbf{x}_1$ and $\mathbf{x}_3$ in right.

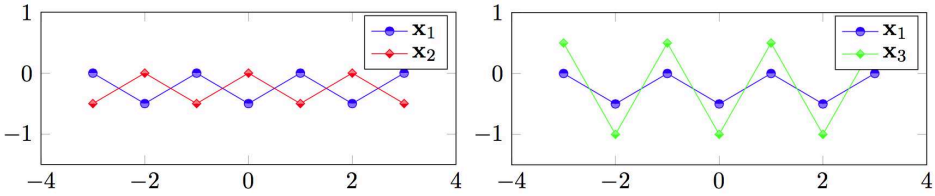


Figure 1.10: Comparison of time series: similar in value, opposite in behavior (left), similar in behavior (right)

Most of the time, two time series sharing same configuration are considered close, even though they have very different values, and vice-versa. Hence, choosing a proper metric can be crucial and very important for the time series comparison. In some cases, several behavior and value-based metrics may be implied. Some propositions show the benefit of involving both behavior and value-based metrics through a combination function. Therefore, to compare the time series and define a proximity measure covering both the behaviors and values components, a weighted linear (or geometric) function combines behavior and value-based metrics. In this context, they could be shown as:

$$\mathcal{BV}_{Lin}(\mathbf{x}_i, \mathbf{x}_j) = \alpha.\mathcal{V}(\mathbf{x}_i, \mathbf{x}_j) \ + \ (1-\alpha).\mathcal{B}(\mathbf{x}_i, \mathbf{x}_j)$$

$$\mathcal{BV}_{Geom}(\mathbf{x}_i, \mathbf{x}_j) = (\mathcal{V}(\mathbf{x}_i, \mathbf{x}_j))^{\alpha} \ . \ (\mathcal{B}(\mathbf{x}_i, \mathbf{x}_j))^{(1-\alpha)}$$

where $\mathcal{V}$ and $\mathcal{B}$ is a value-based and behavior-based metrics, respectively. The parameter $\alpha \in [0, 1]$ is a trade-off between the behavior and value-based components.

On the other hand, a time series may be considered in the spectral representations, which means the time series may be similar because they share the same frequency characteristics. Hence, in some application, the frequential-based metrics (e.g. wavelet transforms, fourier transforms) will be used. More specific works to combine two different metrics through a combination function proposed in [DCN07]; [DCA12].

1.4 Kernels for time series

Over the last ten years estimation and learning methods using kernels have become rather popular to cope with non-linearities, particularly in machine learning. Kernel methods [HSS08] have been proved useful to handle and analyze structured data such as images, graphs and texts. They map the data from the original space (i.e. input space) via a nonlinear mapping function $\Phi(.)$ to a higher dimensional feature space (i.e. Hilbert space), to discover nonlinear patterns (see Figure 1.11). These methods formulate learning problems, in a reproducing kernel Hilbert space, of functions defined on the data domain expanded in terms of a kernel.

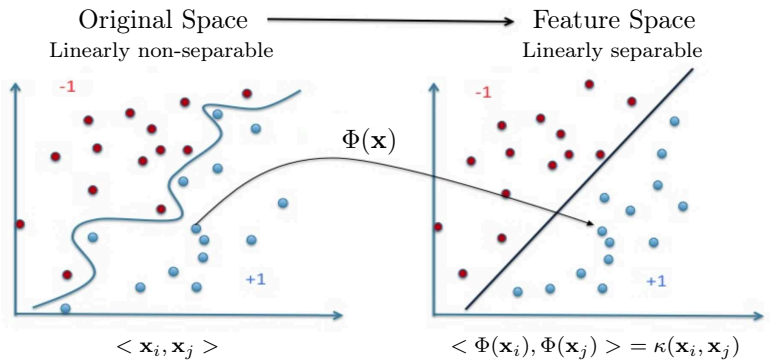


Figure 1.11: Kernel trick: embed data into high dimension space (feature space)

Usually, a kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ is used to directly provide the inner products in a feature space without explicitly defining transformation. The kernel corresponds to the dot product in a (usually high-dimensional) feature space. In this space, the estimation methods are linear, but as long as we can formulate everything in terms of kernel evaluations, we explicitly never have to compute in the high-dimensional feature space. Such inner products can be viewed as measuring the similarity between samples. Let $X = \{x_1, ..., x_N\}$ a set of samples.

Gram matrix (or kernel matrix) of function κ with respect to the $\{x_1, ..., x_N\}$ is a $N \times N$ matrix defined by:

$$K := [\,\kappa(x_i, x_j)\,]_{ij}$$

where κ is a kernel function. If we use algorithms that only depend on the Gram matrix, K, then we never have to know (or compute) the actual features Φ. This is the crucial point of kernel methods.

Positive definite matrix is a real $N \times N$ symmetric matrix, that K_{ij} satisfying:

$$\sum_{i,j} c_i c_j K_{ij} > 0$$

for all $c_i \in \mathbb{R}$.

Positive semidefinite matrix is a real $N \times N$ symmetric matrix, that the Gram matrix K_{ij} satisfying:

$$\sum_{i,j} c_i c_j K_{ij} \geq 0$$

for all $c_i \in \mathbb{R}$. In the following, to simplify, will call them definite. if we really need > 0, we will say **strictly positive definite**.

A symmetric function $\kappa : X \times X \to \mathbb{R}$ which for all $x_i \in X, \forall i \in \{1, ..., N\}$ gives rise to a positive definite Gram matrix is called a **Positive definite kernel**.

For time series, several kernels are proposed in the last years, properties of which will be discussed later. Ideally, a useful kernel for time series should be both positive definite and able to handle time series of structured data. Here, we refer positive definite kernels as kernels. Note that, for simplicity, we have restricted ourselves to the case of real valued kernels. The linear kernel is the simplest kernel function. It is given by the inner product $< \mathbf{x}_i, \mathbf{x}_j >$ plus an optional constant γ. A polynomial kernel allows us to model feature conjunctions up to the order of the polynomial, whereas the Gaussian kernel is an example of Radial Basis Function (RBF) kernel. In Table 1.1, some kernel function examples are given.

Linear Kernel	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + \gamma)$
Polynomial Kernel	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\alpha.\mathbf{x}_i^T \mathbf{x}_j + \gamma)^\delta$
Gaussian Kernel	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(\frac{\|\|\mathbf{x}_i - \mathbf{x}_j\|\|^2}{-2\sigma^2})$

Table 1.1: Kernel function examples

The adjustable parameters σ, α, γ play a major role in the performance of the kernel, and should be carefully tuned to the problem at hand. In the following, different methods to compute the kernel similarity Gram matrix presented.

1.4.1 Temporal correlation coefficient kernel (κ_{CORT})

$\text{CORT}(\mathbf{x}_i, \mathbf{x}_j)$, which is a behavior-based metric between two time series $\mathbf{x}_i$ and $\mathbf{x}_j$ and described in Section 1.3.2.2, is a linear kernel and κ_{CORT} defined as a positive definite linear kernel as following:

$$\kappa_{\text{CORT}}(\mathbf{x}_i, \mathbf{x}_j) = \text{CORT}(\mathbf{x}_i, \mathbf{x}_j) = < \triangle\mathbf{x}_i \ , \ \triangle\mathbf{x}_j >$$

It is easy to prove that $\kappa_{\text{CORT}}(\mathbf{x}_i, \mathbf{x}_j)$ can be constructed from inner products in appropriate Hilbert spaces.

1.4.2 Auto-correlation kernel (κ_{DACO})

To compare the dynamics of two actions Gaidon [GHS11] proposed to compute the distance between their respective auto-correlations. This distance is defined as the Hilbert-Schmidt norm of the difference between auto-correlations detailed in Section 1.3.2.3, and called as associated Gaussian RBF kernel "Difference between Auto-Correlation Operators Kernel". κ_{DACO} is defined by:

$$\kappa_{\text{DACO}}(\mathbf{x}_i, \mathbf{x}_j) = e^{\left[\frac{-1}{\sigma^2}\text{DACO}(\mathbf{x}_i, \mathbf{x}_j)\right]}$$

κ_{DACO} is a positive definite Gaussian kernel, designed specifically for action recognition. Notice that, both κ_{CORT} and κ_{DACO} are temporal kernels under Euclidean alignments. Under time warp, time series should be first synchronized by DTW, then kernels applied. In the following, we introduce the major well-known temporal kernels under dynamic time warping alignment.

1.4.3 Gaussian dynamic time warping kernel (κ_{GDTW})

Following [BHB02], the dynamic time warping distance can be used as a pseudo negative definite kernel to define a pseudo positive definite kernel. κ_{GDTW} is determined by:

$$\kappa_{\text{GDTW}}(\mathbf{x}_i, \mathbf{x}_j) = e^{\left[\frac{-1}{t}\ \text{DTW}(\mathbf{x}_i, \mathbf{x}_j)\right]}$$

and using Sakoe-Chiba constrained dynamic time warping distance, $\kappa_{\text{GDTW}_{sc}}$ is defined by:

$$\kappa_{\text{GDTW}_{sc}}(\mathbf{x}_i, \mathbf{x}_j) = e^{\left[\frac{-1}{t}\ \text{DTW}_{sc}(\mathbf{x}_i, \mathbf{x}_j)\right]}$$

where t is a normalization parameter, and sc is a Sakoe-Chiba band. As the DTW is not a metric (invalid triangle inequality), one could fear that the resulting kernel lacks some necessary properties. Hence, general positive definiteness can not be proven for κ_{GDTW}, as simple counterexamples can be found. However, this kernel can produce good results in some cases [HK02]; [DS02], but a problem of this approach is the quadratic computational complexity [BHB02].

1.4.4 Dynamic time-alignment kernel (κ_{DTAK})

The Dynamic Time-Alignment Kernel (κ_{DTAK}) proposed in [Shi+02] adjusts another similarity measure between two time series by considering the arithmetic mean of the kernel values along the alignment path. κ_{DTAK}, a pseudo positive definite kernel , is then defined by:

$$\kappa_{\text{DTAK}}(\mathbf{x}_i, \mathbf{x}_j) = \max_{\pi \in \mathcal{A}}\ C(\pi)$$

where,

$$C(\pi) = \frac{1}{|\pi|}\sum_{l=1}^{|\pi|}\phi(x_{i\pi_1(l)}, x_{j\pi_2(l)}) = \frac{1}{|\pi|}\sum_{(t,t')\in\pi}^{|\pi|}\phi(x_{it}, x_{jt'})$$

and

$$\phi(x_{it}, x_{jt'}) = \kappa_\sigma(x_{it}, x_{jt'}) = e^{\left[\frac{-1}{\sigma^2}\ \|x_{it} - x_{jt'}\|^2\right]}$$

Dynamic time-alignment kernel function is complicated and difficult to analyze because the input data is a vector sequence with variable length and non-linear time normalization is embedded in the function. It performs DTW in the transformed feature space and finds the optimal path that maximizes the accumulated similarity. Indeed, the local similarity used for the κ_{GDTW} is the Euclidian distance, whereas the one used in κ_{DTAK} is the Gaussian kernel value. κ_{DTAK} is a symmetric kernel function, however, in general, it may still be a non positive definite kernel. Note that, in practice, several ad-hoc methods (e.g. perturb the whole diagonal by the absolute of the smallest eigenvalue) are used to ensure the positive definiteness, when the Gram matrix is not positive definite.

1.4.5 Global alignment kernel (κ_{GA})

In kernel methods, both large and small similarities matter, they all contribute to the Gram matrix. Global Alignment kernel (κ_{GA}) [Cut+07] is not based on an optimal path chosen given a criterion, but takes advantage of all score values spanned by all possible alignments. The κ_{GA} which is positive definite under mild conditions [2], seems to do a better job of quantifying all similarities coherently, because it considers all possible alignments. κ_{GA} is defined by:

$$\kappa_{\mathrm{GA}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{\pi \in \mathcal{A}} \prod_{l=1}^{|\pi|} k(x_{i\pi_1(l)}, x_{j\pi_2(l)})$$

where $k(x, y) = e^{-\lambda \Phi_\sigma(x,y)}$, factor $\lambda > 0$, and

$$\Phi_\sigma(x, y) = \frac{1}{2\sigma^2}\|x - y\|^2 + \log(2 - e^{\frac{-1}{2\sigma^2}\|x-y\|^2})$$

In the sense of kernel κ_{GA}, two sequences are similar not only if they have one single alignment with high score, which results in a small DTW distance, but also share numerous efficient and suitable alignments. Hence, the function Φ_σ is a negative definite kernel, it can be scaled by a factor λ to define a local positive definite kernel $e^{-\lambda\Phi_\sigma}$. Global alignment kernels have been relatively successful in different application fields [JER09]; [RTZ10]; [VS10] and shown to be competitive when compared with other kernels. However, similar to the κ_{GDTW} and κ_{DTAK} kernels, it has quadratic complexity, $O(pT^2)$, where T is the length of time series and p denotes the complexity of the measure between aligned instants.

1.4.6 Triangular global alignment kernel (κ_{TGA})

The idea of Sakoe-Chiba [SC78], to speed up the computation of DTW, applied to the κ_{GA} to introduce fast global alignment kernels, named as Triangular Global Alignment kernels. κ_{TGA} kernel [Cut11] considers a smaller subset of such alignments rather than κ_{GA}. It is faster to compute and positive definite, and can be seen as trade-off between the full κ_{GA} (accurate but slow) and a Sakoe-Chiba Gaussian kernel (fast but limited). This kernel is defined by:

$$\kappa_{\mathrm{TGA}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{\pi \in \mathcal{A}} \prod_{l=1}^{|\pi|} k(x_{i\pi_1(l)}, x_{j\pi_2(l)})$$

$$k(x_{i\pi_1(l)}, x_{j\pi_2(l)}) = \frac{w_{\pi_1(l),\pi_2(l)} k_\sigma(x_{i\pi_1(l)}, x_{j\pi_2(l)})}{2 - w_{\pi_1(l),\pi_2(l)} k_\sigma(x_{i\pi_1(l)}, x_{j\pi_2(l)})}$$

where, w is a radial basis kernel in $\mathbb{R}$ (a triangular kernel for integers):

$$w(t, t') = \left(1 - \frac{|t - t'|}{c}\right)$$

[2] κ_{GA} is positive definite, if $\kappa/(1 + \kappa)$ is positive definite.

which guarantees that only the alignments π that are close to the diagonal are considered and the parameters c is the Sakoe-Chiba bandwidth. Note that as c increases the κ_{TGA} converges to the κ_{GA}.

In fact, many generic kernels (e.g. Gaussian kernels), as well as very specific ones, describe different notions of similarity between time series, which do not correspond to any intuitive or easily interpretable high-dimensional representation. They can be used to compute similarities (or distances) in feature spaces using the mapping functions. Precisely, the class of kernels that can be used is larger than those commonly used in the kernel methods which known as positive definite kernels.

1.5 Conclusion

A large variety of real world applications, such as meteorology, geophysics and astrophysics, collect observations that can be represented as time series. Time series data mining and time series analysis can be exploited from research areas dealing with sequences and signals, such as pattern recognition, image and signal processing. The main problem that arises during time series analysis is comparison of time series.

There has been active research on quantification of the "similarity", the "dissimilarity" or the "distance" between time series. Even, looking for the patterns and dependencies in the visualizations of time series can be a very exhausting task and the aim of this work is to find the more similar behavior time series, while the value-based proximity measures are the most studied approaches to compare the time series. Most of the time, two time series sharing same configuration are considered close, even they have very different values, their appearances are similar in terms of form. In both comparison metrics, a crucial question is establishing what we mean by "similar" or "dissimilar" time series due to the dynamic of the series (with or without considering delay).

To handle and analysis non-linearly structured data, a simple way is to treat the time series as vectors and simply employ a linear kernel or Radial basis function kernel. For this, different kernels have been proposed. To discover non-linear patterns, they map the data from the original input space to a higher dimensional feature space, called Hilbert space. They can be used to compute similarity between the time series. Ideally, a useful kernel for time series should be positive definite and able to handle the temporal data. Notice that, the computational complexity of kernel construction and evaluation can play a critical role in applying kernel methods to time series data.

In summary, finding a suitable proximity measure is a crucial aspect when dealing with the time series (e.g. a kernel similarity, a similarity or dissimilarity measure, a distance), that captures the essence of the time series according to the domain of application. For example, Euclidean distance is commonly used due to its computational efficiency; however, it is very brittle for time series and small shifts of one time series can result in huge distance changes. Therefore, more sophisticated distances have been devised to be more robust to small

fluctuations of the input time series. Notably, Dynamic Time Warping (DTW) has enjoyed success in many areas where its time complexity is not an issue. Using the discussed distances and (dis)similarity proximity measures, we will be able to compare time series and analyze them. This can provide useful insights on the time series, before the averaging and centroid estimation step. In the next chapter, we will discuss about averaging time series problems, difficulties and complexities.

Chapter 2

Time series averaging and centroid estimation

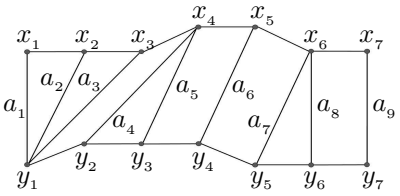
Sommaire

Averaging a set of time series, under the frequently used dynamic time warping, needs to address the problem of multiple temporal alignments, a challenging issue in various domains. Under temporal metrics, one standard way to average two time series is to synchronize them and average each pairwise temporal warping alignment. To average more than two time series, the problem becomes more complex as one needs to determine a multiple alignment that link simultaneously all the time series on their commonly shared similar elements. In this chapter, we present initially the definition of a consensus sequence and multiple temporal alignment problems, then we review major progressive and iterative centroid estimation approaches under time warp, and discuss their properties and complexities.

2.1 Introduction

Estimating the centroid of a set of time series is an essential task of many data analysis and mining processes, as summarizing a set of time series, extracting temporal prototypes, or clustering time series. Averaging a set of time series, under the frequently used dynamic time warping [KL83]; [SK83] metric or its variants [Ita75]; [Rab89]; [SC78], needs to address the tricky multiple temporal alignments problem, a challenging issue in various domains.

Under time warp, one standard way to estimate the centroid of two time series is first to synchronize them (to find the pairwise links), and then average each pairwise temporal warping alignment. In this way, the centroid can be estimated as the average of the linked elements as illustrated in Figure 2.1.



i		1	2	...	9
	x	x_1	x_2	...	x_7
	y	y_1	y_1	...	y_7
	c	avg(x_1, y_1)	avg(x_2, y_1)	...	avg(x_7, y_7)

Figure 2.1: Pairwise temporal warping alignment between **x** and **y** (left), the estimation of the centroid **c** of time series **x** and time series **y** in the embedding Euclidean space (right). A dimension i in the embedded Euclidean space identifies a pairwise link a_i.

Averaging more than two time series under time warp is a very complex problem, because we need to determine a multiple alignment that links simultaneously all the time series on their commonly shared similar elements. That multiple alignment defines, similarly, a new Euclidean embedding space where time series can be projected and the global centroid estimated. Note that each dimension identifies, this time, a hyper link that connects more than two elements. Finding the multiple temporal alignment of a set of time series or its average sequence is a chicken-or-egg problem: knowing the average sequence under time warp provides a multiple temporal alignment and vice-versa.

A first direct approach to determine a multiple alignment is to search, by dynamic programming, the optimal path within an N-dimensional grid that crosses the N time series. The complexity of this approach prevents its use, as it constitutes an NP-complete problem with a complexity of $O(T^N)$ that increases exponentially with the number of time series N and the time series length T [CL88]; [Jus01]. A second manner, that characterizes progressive approaches, is based on combining progressively pairs of time series centroids to estimate the global centroid. Progressive approaches may however suffer from the error propagation problem: Early errors propagate in all subsequent centroids through the pairwise combinations. The third approach is iterative. It works similarly to the progressive approach but reduces the error propagation by repeatedly refining the centroid and realigning it to the initial time series. The progressive and iterative approaches for centroid estimation are mainly derived from the multiple sequence alignment methods to address the challenging problem of aligning more than two time series [CNH00]; [SLY06]; [THG94]. In general, the main progressive and iterative approaches are of heuristic nature and are limited to the dynamic time warping metric [SKDCG16a]. Here, we first introduce the definition of pairwise time series averaging, prior to study the consensus sequence. Next, we review some major progressive and iterative approaches for time series averaging under the dynamic time warping to handle the tricky multiple temporal alignment problem.

Definition 2.1
Under the dynamic time warping, given $\mathbf{x}_i$ *and* $\mathbf{x}_j$ *and their dynamic time warping alignment* $\boldsymbol{\pi}^*$*, the centroid* $\mathbf{c}(\mathbf{x}_i, \mathbf{x}_j)$ *is defined by averaging their aligned elements as follows:*

$$\mathbf{c}(\mathbf{x}_i, \mathbf{x}_j) = (Avg(x_{i\pi_1^*(1)}, x_{j\pi_2^*(1)}), ..., Avg(x_{i\pi_1^*(m)}, x_{j\pi_2^*(m)}))$$

where Avg is a standard numerical averaging function.

Figure 2.2 illustrates the pairwise time series centroid estimation under the dynamic time warping. Note that the length $T \leq m \leq 2T - 1$ of the centroid $\mathbf{c}(\mathbf{x}_i, \mathbf{x}_j)$ is generally higher than the length of the centered time series.

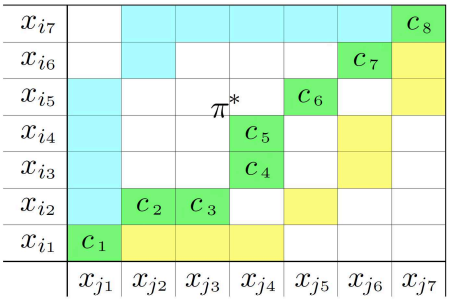


Figure 2.2: Three possible alignments (paths) are displayed between $\mathbf{x}_i$ and $\mathbf{x}_j$, $\boldsymbol{\pi}^*$ (the green one) being the dynamic time warping one. The centroid $\mathbf{c}(\mathbf{x}_i, \mathbf{x}_j) = (c_1, ..., c_8)$ is defined as the average of the linked elements through $\boldsymbol{\pi}^*$, with for instance $c_3 = Avg(x_{i2}, x_{j3})$.

2.2 Consensus sequence

The time series comparison is a key point for many problems, as finding a consensus sequence in molecular biology, bioinformatics or data mining. In the context of sequences, this term is used with two meanings: i) the medoid sequence, ii) the average sequence of the set of time series.

2.2.1 Medoid sequence

The concept refers to a formal definition, corresponding to a sequence in the center of a set of time series. A medoid is a representative time series in a set that minimizes the sum of distances (i.e. inertia) to all other time series within the same set. Figure 2.3 shows a medoid as a consensus between some time series sequences.

In the context of (un)supervised learning, many algorithms require a method to represent information from a set of time series in one and only one sequence. Algorithms like k-medoids

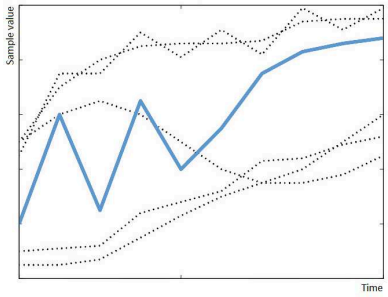


Figure 2.3: Medoid as a prototype

are using the medoid of a set of objects, which is better in handling noises and outliers than k-means. But, mostly in other algorithms we need to average a set of time series as an optimal consensus.

2.2.2 Average sequence

Since our purpose is to define an average sequence minimizing the sum of distances to all time series of a set (i.e. inertia), we give the definition of an average sequence when the corresponding distance is DTW.

Definition 2.2
Let $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ *a set of time series* $\mathbf{x}_i = (x_{i1}, ..., x_{iT})$ $i \in \{1, ..., N\}$, *and* $\mathbb{S}$ *is the space of all time series sequences.* $\forall \mathbf{s} \in \mathbb{S}$, *the average sequence* $\mathbf{c}$ *should satisfy the following:*

$$\sum_{i=1}^{N} \mathbf{DTW}(\mathbf{x}_i, \mathbf{c}) \leq \sum_{i=1}^{N} \mathbf{DTW}(\mathbf{x}_i, \mathbf{s})$$

Since no information on the length of the average sequence $\mathbf{c}$ is available, the search cannot be limited to sequences of a given length, so all possible length values for averages have to be considered. In the following, we will review some techniques used to determine the multiple temporal alignment (and average) of a set of time series.

2.3 Multiple temporal alignments

Multiple temporal alignment of time series is an important problem with applications in many scientific domains such as speech recognition [RJ93], computer graphics [BW95], astronomy [Lis+05], computer vision [CI02], and bio-informatics [AC01]. The multiple temporal alignment is computable by extending DTW for aligning N time series. This idea can be generalized by computing DTW in a N-dimensional hypercube. Given this global alignment, the consensus can be found by averaging column by column of the multiple alignments. Thus, the first direct approach to determine a multiple temporal alignment is to search, by dynamic programming, the optimal path within an N-dimensional grid that crosses the N time series.

2.3.1 Dynamic programming

Dynamic programming is a direct method to determine a multiple temporal alignment by search the optimal path within an N-dimensional grid that crosses the N time series. The complexity of this approach prevents its use, as it constitutes an NP-complete problem with a complexity of $O(T^N)$ that increases exponentially with N the number of time series, and T the time series length [CL88]; [Jus01].

The progressive and iterative approaches for averaging a set of time series are mostly derived from the multiple sequence alignment methods to address the challenging problem of aligning more than two time series [CNH00]; [SLY06]; [THG94]. To estimate the centroid of more than two time series, several heuristic approaches have been proposed. In the following, we review the main studies related to time series multiple temporal alignments as well as the major progressive and iterative approaches for time series averaging under time warp.

2.3.2 Progressive approaches

The progressive approaches estimate the global centroid $\mathbf{c}$ by combining pairwise time series centroids through different strategies. Here, we present the most well-known ones.

2.3.2.1 NonLinear Alignment and Averaging Filters (NLAAF)

In the past decades, many averaging appraoches have been introduced, but only a few of them have been adapted to time series averaging, mining and clustering. For instance, Gupta et al. in [GDMS96], proposed a time series averaging method based on a tournament scheme, called "*NonLinear Alignment and Averaging Filters* (NLAAF)". First, pairs of time series are selected randomly, and then aligned according to the DTW. That way, $(N/2)$ [1] averaged sequences are created. The same process is iterated on the estimated centroids , until only one sequence is obtained as a global centroid. In this approach, the averaging method between two time series is applied $(N-1)$ times, as illustrated in Figure 2.4, where $\mathbf{c}(\mathbf{x}_i, \mathbf{x}_j)$ refers to the estimated centroid of time series $\mathbf{x}_i$ and time series $\mathbf{x}_j$.

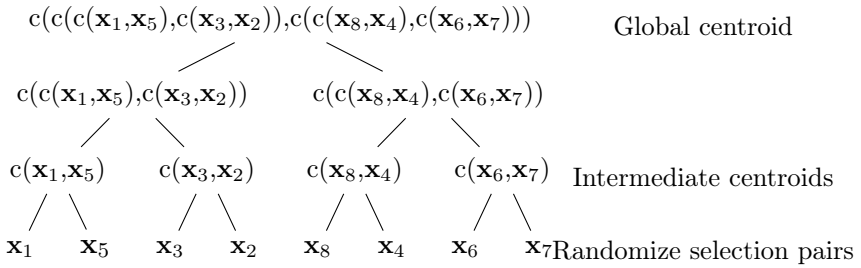


Figure 2.4: Centroid estimation by random pairwise centroid combination.

[1] N denotes the number of total times series to be averaged.

In this method, each element of a centroid is computed as the mean of each linked elements in the DTW alignment. The major drawback of NLAAF lies in the growth of its resulting length, because each use of the averaging method can almost double the length of the average sequence. As classical datasets comprise thousands of time series, with each one including hundreds of data points, simply storing the resulting average may be impossible. This length problem is moreover worsened by the quadratic computational complexity of DTW, that grows bi-linearly with the lengths of the sequences. That is why NLAAF method is generally used in conjunction with a process reducing length of the average, unfortunately leading to information loss and unsatisfactory approximation. Additionally, the average strongly depends on the random selection of time series and different choices lead to different results.

2.3.2.2 Prioritized Shape Averaging (PSA)

To avoid the bias induced by random selection, Niennattrakul et al. among others [THG94]; [NR07]; [NR09] proposed a framework of shape averaging called "*Prioritized Shape Averaging* (PSA)" based on the hierarchical clustering. The pairwaise time series centering is guided by the dendrogram obtained through the hierarchical clustering.

The PSA uses hierarchical clustering as a manner to identify priorities between time series. In particular, to estimate the global centroid, the set is first clustered using the agglomerative clustering to get a hierarchical relationship among the whole time series. The simple linkage or complete linkage is considered in general to fasten the dendrogram build, where almost the average linkage or centroids are the best-performed methods in result. Subsequently, the pairwise time series centroids are combined respectively to their clustering order in the dendrogram (see Figure 2.5).

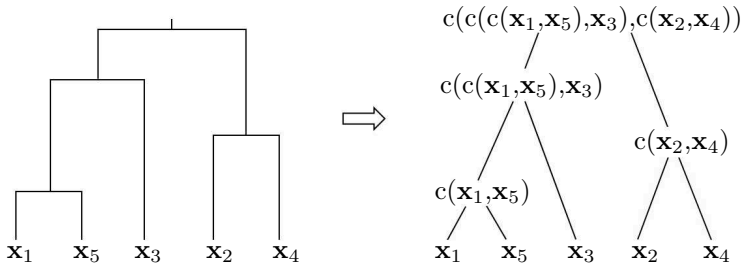


Figure 2.5: Centroid estimation by centroid combination through clustering dendrogram.

Each parent node is averaged in a bottom-up manner using a weighted DTW averaging, called SDTW [2]. So, the most similar time series are averaged first. Note that the weight of an averaged sequence is calculated from the number of the time series upon which the averaged sequence is formed. Initially, all the time series have the same weight of one.

[2]Scaled Dynamic Time Warping

An example of averaging six sample time series using PSA is described in Figure 2.6. According to the dendrogram obtained thorough hierarchical clustering, first the time series $\mathbf{x}_2$ and $\mathbf{x}_3$ are averaged. The average sequence denoted by $\mathbf{c}(\mathbf{x}_2, \mathbf{x}_3)$, has the weight of two. Then, the intermediate centroid $\mathbf{c}(\mathbf{x}_1, \mathbf{c}(\mathbf{x}_2, \mathbf{x}_3))$ is computed by averaging the time series $\mathbf{x}_1$ and the average sequence $\mathbf{c}(\mathbf{x}_2, \mathbf{x}_3)$. The intermediate centroid $\mathbf{c}(\mathbf{x}_1, \mathbf{c}(\mathbf{x}_2, \mathbf{x}_3))$ will have the weight of three, since the time series $\mathbf{x}_1$ and $\mathbf{c}(\mathbf{x}_2, \mathbf{x}_3)$ have weight of one and two, respectively. The process continues till one obtains a global centroid.

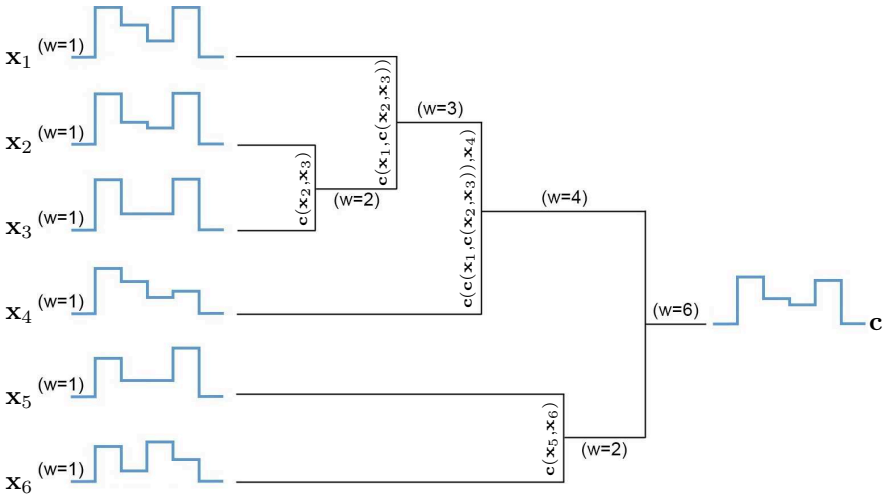


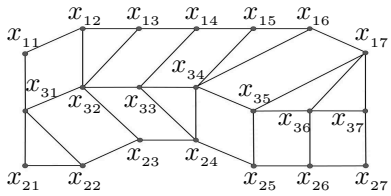
Figure 2.6: Example of six time series sequence averaging using PSA

Although this hierarchical averaging approach aims to remove the bias induced by random selection, the growth length of the average sequence remains a problem. Additionally, local averaging strategies like NLAAF or PSA may let an initial approximation error propagate throughout the averaging process. If the averaging process has to be repeated (e.g. during k-means clustering iterations), the effects may dramatically alter the quality of the result. This is why a global averaging approach is desirable, where time series would be averaged all together, with no sensitivity to their order of consideration.

2.3.2.3 Cross-Words Reference Template (CWRT)

A direct way to estimate the centroid is proposed by Abdulla [ACS03], where a dynamic time warping between each time series and a reference one, generally the time series medoid, is first performed. Each time series is then described in a representation space defined by the reference medoid by resampling, stretching and shortening operations (see Figure 2.7). Finally the global centroid $\mathbf{c}$ is computed by averaging the time-aligned time series across each point. The approach is called "*Cross-Words Reference Template* (CWRT)".

The global estimated centroid $\mathbf{c}$ has the same length as the reference time series (e.g. medoid), and the result does not depend on the order in which time series are processed. But the method is a heuristic approach, with no guarantee of optimality.



	1	2	...	7
$\mathbf{x}_1$	x_{11}	x_{12}, x_{13}	...	x_{17}
$\mathbf{x}_2$	x_{21}, x_{22}	x_{23}	...	x_{27}
$\mathbf{x}_3$	x_{31}	x_{32}	...	x_{37}
$\mathbf{c}$	avg$(x_{11}, x_{21}, x_{22}, x_{31})$	avg$(x_{12}, x_{13}, x_{23}, x_{32})$	...	avg(x_{17}, x_{27}, x_{37})

Figure 2.7: Centroid estimation based on a reference time series. The DTW is performed between time series $\mathbf{x}_1$, $\mathbf{x}_2$ and the reference time series $\mathbf{x}_3$ (left). Time series $\mathbf{x}_1$ and $\mathbf{x}_2$ are embedded in the space defined by $\mathbf{x}_3$ (right) where the global centroid is estimated, and 'avg' is the standard mean function.

2.3.3 Iterative approaches

The progressive approaches described above suffer from the early error propagation through the set of pairwise centering combinations. To avoid that, iterative approaches proceed similarly to the progressive ones, but reduce the error propagation by repeatedly refining the centroid and realigning it to the initial time series, until its stabilization [HNF08]; [PKG11].

2.3.3.1 Dtw Barycenter Averaging (DBA)

Petitjean et al. in [PKG11] proposed a global averaging method, called "*Dtw Barycenter Averaging* (DBA)". The method consists in iteratively refining an initially average sequence (potentially arbitrary), in order to minimize its distance to the set of time series. The aim is to minimize inertia as the sum of squared DTW distances from the average sequence to the set of time series. Technically, for each refinement iteration DBA works in two steps:

- Computing the DTW between each time series and the temporary average sequence to find the temporal alignments.
- Updating each element of the average sequence with the barycenter of the elements aligned to it during the first step.

Figure 2.8 shows three iterations of DBA on an example with two time series $\mathbf{x}_i$ and $\mathbf{x}_j$, while $\mathbf{c}$ is the global estimated centroid.

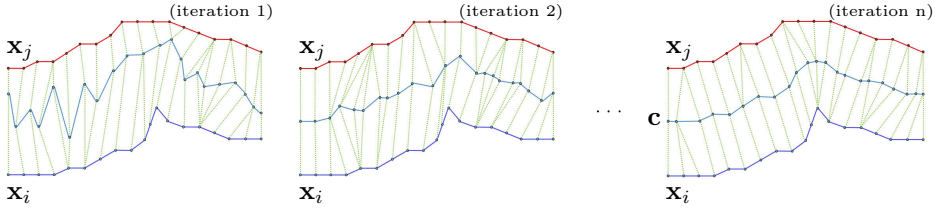


Figure 2.8: DBA iteratively adjusting the average of two time series $\mathbf{x}_i$ and $\mathbf{x}_j$

In a nutshell, the DBA under time warp is a global approach that can average a set of time series all together. The global estimated centroid has the same length as the initial average sequence, and like CWRT, the result is not depending on the order of time series. However the time complexity of DBA approach is smaller than NLAAF and PSA [PKG11], but the time complexity problem remains.

All the progressive and iterative methods discussed in this chapter are heuristic, with no guarantee of optimality [SKDCG15]; [SKDCG16a]. Lastly, if the provided approximations are accurate for globally similar time series, they are in general poor for time series which share local characteristics with distinctive global behaviors. To circumvent these problems, we proposed in [SKDCG16b] a tractable fast centroid estimation that captures both global and local temporal features under time warp measures. It formalizes the multiple time series averaging problem as an optimization problem and proposes a solution yielding a local optimum. We will discuss the details of that in next sections.

2.4 Conclusion

We have seen that most of the works on averaging a set of time series can be analyzed along two dimensions: first, the way they consider the individual time series for averaging, and second, the way they compute the elements of the resulting average sequences. These two characteristics have proved useful to classify the existing time series averaging techniques. They are also useful to expand new solutions. The main shortcoming of most existing methods is their use of pairwise averaging. To compute the mean of N time series by pairwise averaging, the order of time series influences the quality of the result. This is why a global averaging approach is desirable, with no sensitivity to their order of consideration.

Let us summarize the main characteristics of the described above approaches. In both NLAAF and PSA, the length of the global centroid increases with the number of time series to be averaged. The length of the centroids estimated by CWRT or DBA is however the same as the reference time series length. Furthermore, all the progressive and iterative approaches are heuristic, with no guarantee of optimality. Lastly, even if the provided approximations are accurate for globally similar time series, they are in general poor for time series that share local characteristics with distinctive global behaviors.

In next chapter, we present the k-means clustering approach, the most popular clustering algorithm, and its variations, prior to introduce the proposed fast and accurate centroid estimation approach under time warp to capture both global and local temporal features and its application in clustering.

Chapter 3

Generalized k-means for time series under time warp measures

Sommaire

Earlier, the clustering techniques have been extensively applied in various domains such as pattern recognition, signal processing, biology, and so forth. k-means based clustering and all its variations, is among the most popular clustering algorithms. To generalize k-means for temporal data, centroid estimation is a key issue because it requires aligning multiple temporal data simultaneously. In addition, one needs to consider a suitable time warp metric, capable to extract both global and local characteristics of time series. But how can we define a good metric? How can we use it in a proper way for centroid estimation ? and what is its applicable usage in clustering?
In this chapter, the above questions are addressed through the following points: i) an extension of the standard time warp measures to consider both global and local temporal differences, ii) a tractable and fast estimation of the cluster representatives based on the extended time warp measures, and iii) a generalization of k-means clustering for temporal data under the extended time warp measures.

3.1 Clustering: An introduction

Clustering is a fundamental task in data mining and can be considered as one of the most important unsupervised learning problems. There is an increasing interest in the use of clustering methods in machine learning, pattern recognition, image processing and information retrieval. It has also a rich history in other disciplines such as biology, psychology, archaeology, geology and marketing. The goal of clustering is to partition data points into homogeneous groups (or clusters) in such a manner that similar data points are grouped together, while different data points belong to different groups. Formally, the clustering structure is represented as a set of clusters $C = \{C_1, ..., C_k\}$ of X, the set of data, such that:

$$\bigcup_{i=1}^{k} C_i = X$$

and for all $i \neq j$;

$$C_i \bigcap C_j = \emptyset$$

The clustering methods can be classified according to: i) the type of input data to the algorithm, ii) the clustering criteria defining the similarity, dissimilarity or distance between data points, and iii) the theory and fundamental concepts. Consequently many clustering algorithms have been proposed in the literature, each one uses a different scientific discipline. Fraley and Raftery [FR98] suggest dividing the clustering algorithms into two main groups: hierarchical and partitioning. Han and Kamber [HK01] propose categorizing the algorithms into additional three main categories: density-based, model-based clustering and grid-based methods. Hierarchical methods make the clusters by recursively partitioning the data points in either a top-down or bottom-up manner. For example, in agglomerative hierarchical clustering, each data point initially represents a cluster of its own. Then clusters are merged, according to some similarity measure, until the desired cluster structure is obtained. The result of this clustering method is a dendrogram. Density-based methods assume the data points that belong to each cluster are drawn from a specific probability distribution [DE93]. The idea is to continue growing the given cluster as long as the density (or the number of data points) in the neighborhood exceeds some pre-defined threshold. The density-based methods are designed for discovering clusters of arbitrary shape which are not necessarily convex. Model-based clustering methods attempt to optimize the fit between the given data and some mathematical models. These methods find characteristic descriptions for each group. The most frequently used model-based clustering methods are decision trees and neural networks. Grid-based methods partition the space into a finite number of cells that form a grid structure. All of the operations for clustering are performed on the grid structure. The grid-based have the fastest processing time that typically depends on the number of the grid instead of the data points [HK01].

In this thesis, we mainly focus on partitioning methods, which use an iterative way to create the clusters by moving data points from one cluster to another, based on a distance

measure, starting from an initial partitioning. Such clustering methods typically require that the number of clusters k will be pre-set by the user (e.g. elbow method[1], see Figure 3.1).

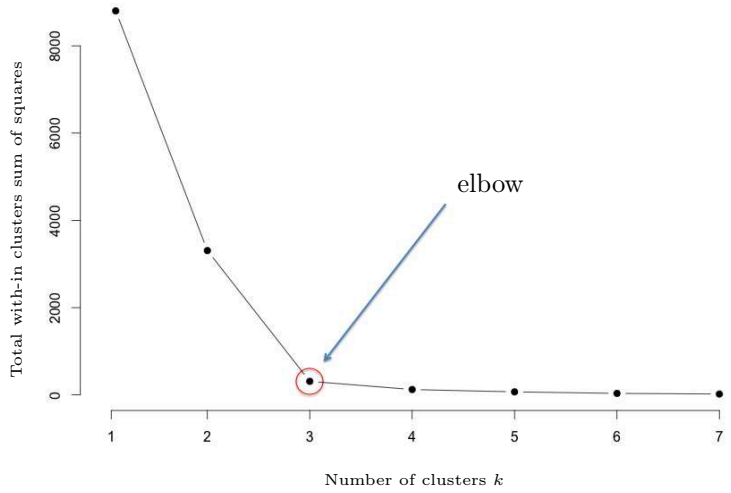


Figure 3.1: The elbow method suggests k=3 cluster solutions

k-means clustering is among the most popular clustering algorithms, as it provides a good trade-off between quality of the solution obtained and its computational complexity [AV07]. In the following, we review the major k-means based clustering algorithms and their formalization, difficulties and complexities.

k-means

k-means algorithm is one of the most popular and simplest unsupervised clustering techniques and it is commonly used in medical imaging, bio-metrics, computer vision, pattern recognition and related fields. Even though k-means was first proposed over 50 years ago [BH65]; [Mac67], it is still one of the most widely used algorithms for clustering.

Generally, k-means is a clustering method that aims to find k centroids, one for each cluster, that minimize the sum of distance of each data point from its respective cluster centroid. It solves, for $\mathbf{x}_i \in \mathbf{X}$:

$$\underset{\{\mathbf{C}_1,\ldots,\mathbf{C}_k\}}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \mathbf{C}_j} d(\mathbf{x}_i, \mathbf{c}_j) \tag{3.1}$$

where $(\mathbf{C}_1, ..., \mathbf{C}_k)$ are k non-overlapping clusters, $\mathbf{c}_j$ is the representative of cluster $\mathbf{C}_j$, and d is a distance function (e.g., Euclidean, DTW).

[1]The idea is this: run k-means clustering on the dataset for a range of values of k (start with $k = 1$), and for each value of k compute the inertia as the sum of distances between each member of a cluster and its centroid. Then, plot a line chart of this inertia for each value of k. If the line chart looks like an arm, then the "elbow" on the arm is the value of k that is the best.

Algorithm 1 k-means clustering $(\mathbf{X}, k)$

Input: $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)$
Input: k the number of clusters
Output: $\{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_k\}$ (set of cluster centroids)
$p = 0$
Randomly choose k objects and make them as initial centroids $(\{\mathbf{c}_1^{(0)}, \mathbf{c}_2^{(0)}, ..., \mathbf{c}_k^{(0)}\})$
repeat
  Assign each data point to the cluster with the nearest centroid
  $p \leftarrow p + 1$
  // **Centroid update**
  for j:=1 to k **do**
    Update the centroid $\mathbf{c}_j^{(p)}$ of each cluster using Eq. 3.2
  end for
until $\mathbf{c}_j^{(p)} \approx \mathbf{c}_j^{(p-1)}$ $\forall j = 1, 2, ..., k$
Return $\mathbf{c}_1$, $\mathbf{c}_2$, ... , $\mathbf{c}_k$

The k-means clustering is presented in Algorithm 1. To initialize the k-means, one needs to specify the number of clusters k. The algorithm starts with an initial set of cluster centers (i.e. centroids) $\{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_k\}$, chosen at random or according to some heuristic procedure. The clustering algorithm uses an iterative refinement technique. In each iteration, each data point is assigned to its nearest cluster centroid. Then the cluster centroids are re-calculated. The centroid $\mathbf{c}$, the center of each cluster, is calculated as the mean of all the data points belonging to that cluster:

$$\mathbf{c}_j = \frac{1}{N_j} \sum_{\mathbf{x}_i \in cluster_j} \mathbf{x}_i \tag{3.2}$$

where $N_j = |cluster_j|$ is the number of data points belonging to cluster j.

The refinement steps are repeated until the centroids no longer move. The complexity of each iteration of the k-means algorithm performed on N data points is $O(k \times N)$. This linear complexity is one of the reasons for the popularity of the k-means clustering algorithms. Even if the number of data points is substantially large, this algorithm is computationally attractive. Other reasons for the $k-$means clustering algorithm's popularity are simplicity of implementation and speed of convergence. A proof of the finite convergence of the k-means algorithms is given in [SI84]. However, it is shown that under certain conditions the algorithm may fail to converge to a local minimum.

The standard k-means algorithm has some limitations. Firstly, the empty clusters can be obtained if no data points are allocated to a cluster during the assignment step. There exists several strategies to handle the empty clusters; such as choosing a data point from the cluster with the highest within inertia, instead of the centroid of the empty cluster. Secondly, the solution depends heavily on the initial cluster centroids, since the k-means cost function is non-convex. Sometimes different initializations lead to very different final clustering results

(see Figure 3.2). A simple but very popular solution for this problem is the use of multiple runs (restarts), where the clusters centroids are randomly placed at different initial positions, hence better local minima can be found. Still we have to decide on the number of restarts and also we are never sure if the initializations are sufficient or not. In addition, the algorithm is sensitive to the noises and outliers. Data points with extremely large values as outliers may substantially distort the distribution of the data. k-medoids clustering which is more robust than k-means in the presence of outliers will be discuss in the following.

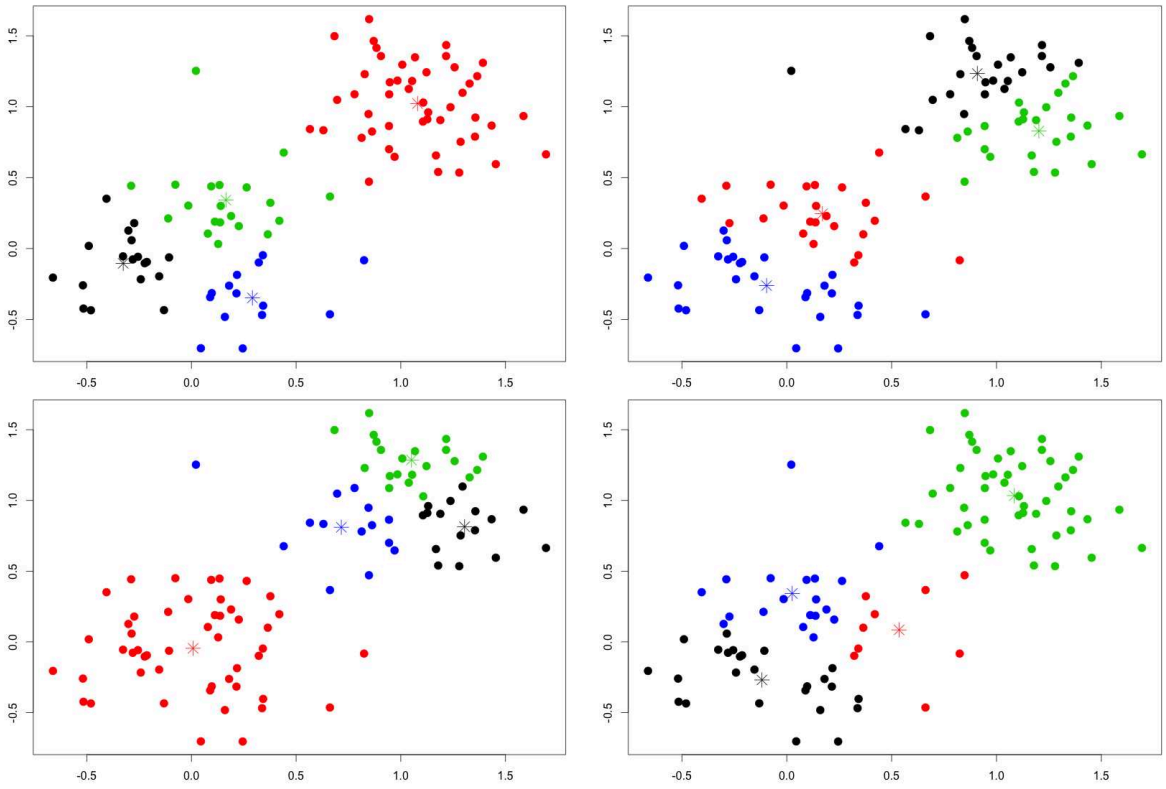


Figure 3.2: k-means: different initializations lead to different clustering results

k-medoids

k-medoids clustering [KR87] is very similar to the k-means clustering algorithm. The major difference between them is that: while a cluster is represented with its center in the k-means algorithm, it is represented with the most centrally located data point in a cluster (i.e. medoid) in the k-medoids clustering. k-medoids uses representative data points as reference points instead of taking the mean value of the data points in each cluster [Zho+05]. The k-medoid minimizes the sum of distance of each data point from its respective cluster medoid, which solves:

$$\underset{\{\mathbf{C}_1,\ldots,\mathbf{C}_k\}}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \mathbf{C}_j} d(\mathbf{x}_i, \mathbf{m}_j) \tag{3.3}$$

where $(\mathbf{C}_1, ..., \mathbf{C}_k)$ are k non-overlapping clusters, $\mathbf{m}_j$ is the medoid of cluster $\mathbf{C}_j$, and d is a distance function.

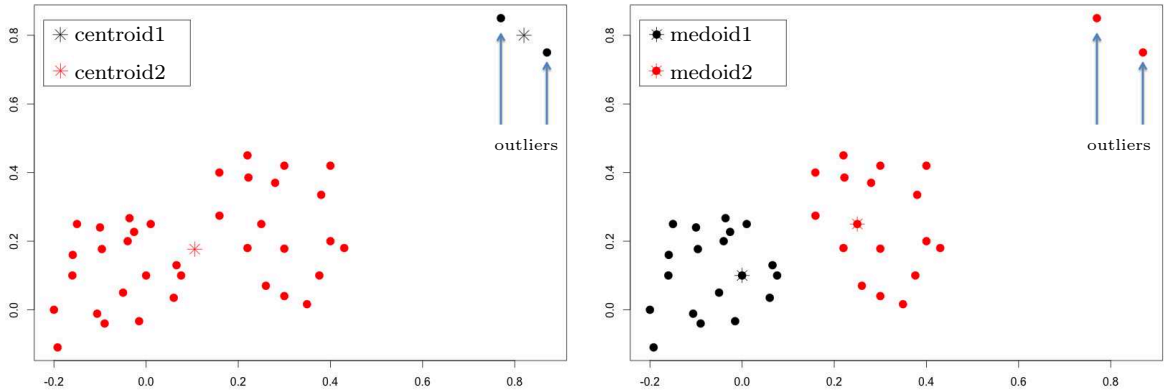


Figure 3.3: Outliers effect: k-means clustering (left) vs. k-medoids clustering (right)

Similarly to the k-means, the k-medoids clustering also requires to specify k, number of desired clusters, in advance. Output results and runtime depend upon initial centroids for both of these clustering methods. As we mentioned before, the k-medoids clustering is less sensitive to noisy data and outliers than the k-means, because a medoid is less influenced by noises, outliers or other extreme values than a mean (see Figure 3.3). But in general, k-medoids computationally are much costlier than the k-means clustering. The complexity of each iteration is $O(k \times (N-k)^2)$, where N is number of data points and k is number of clusters. In a simple way, the comparison between k-means and k-medoids illustrated in Table 3.1.

k-means	k-medoids
complexity per iteration $O(k \times N)$	complexity per iteration $O(k \times (N-k)^2)$
sensitive to noises and outliers	less sensitive to noises and outliers
implementation of algorithm is easy	implementation of algorithm is complicated
require to specify k	require to specify k
comparatively more efficient	comparatively less efficient

Table 3.1: Comparison between k-means and k-medoids

The most common realization of k-medoid is Partitioning Around Medoids (PAM) [KR90], which is known to be the most powerful. The algorithm has two phases: i) in the first phase, "*Build*", a collection of k data points are selected for an initial set of cluster medoids, ii) in the second phase, "*Swap*", one tries to improve the quality of clustering by exchanging the

selected data points with non-selected data points. PAM clustering works effectively for small datasets, but does not scale well for large datasets due to its complexity [NH94]; [JHT01]. To deal with very large datasets, a sampling-based method, called CLARA (Clustering LARge Applications) can be used. The CLARA chooses multiple samples of data, applies the PAM clustering on each sample, and returns its best clustering as output [KR90]. Finally, the choice of clustering algorithm depends both on the type of data and on the particular purpose and application.

Kernel k-means

A major drawback to k-means clustering algorithm is that it does not work for non-linearly separable clusters (see Figure 3.4). Kernel-based clustering techniques address this limitation by introducing a kernel function to capture the non-linear structure in data.

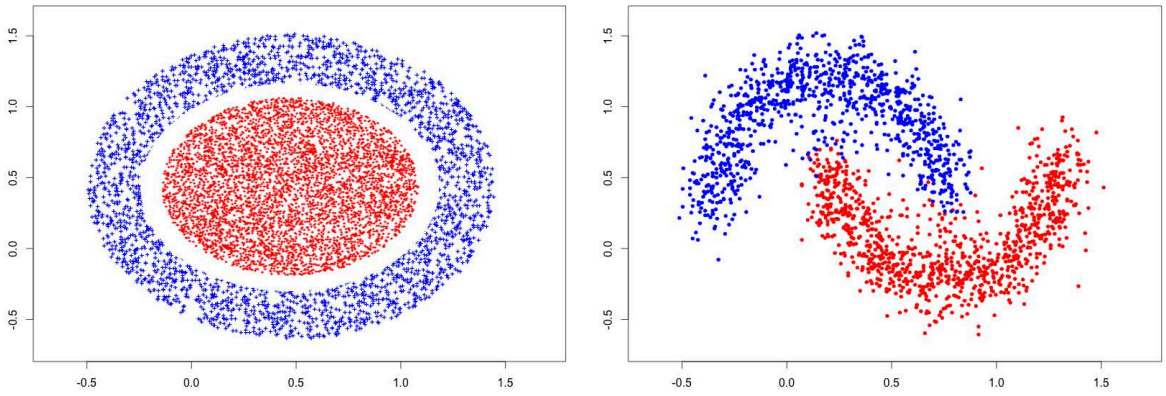


Figure 3.4: Non-linearly separable clusters

Kernel k-means is an extension of standard k-means algorithm that maps data points from the input space to a higher dimensional feature space through a non-linear transformation Φ and minimizes the clustering error in the feature space [DWK05]. When k-means is applied in this feature space, the linear separators in the feature space correspond to nonlinear separators in the input space. So, separation and clustering may be easier in higher dimensional feature space (see Figure 3.5).

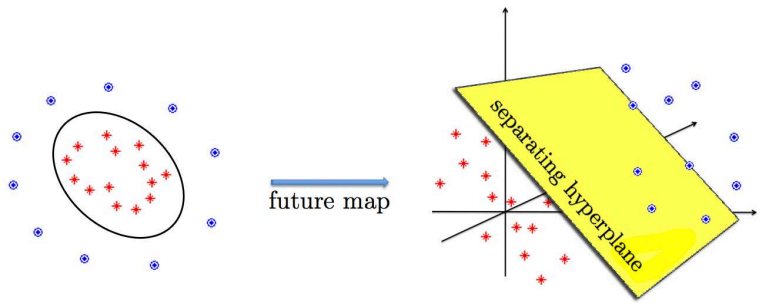


Figure 3.5: The kernel trick - complex in low dimension (left), simple in higher dimension (right)

Figure 3.6 shows the clustering result comparison between the k-means vs. the kernel k-means clustering for a sample non-linear dataset.

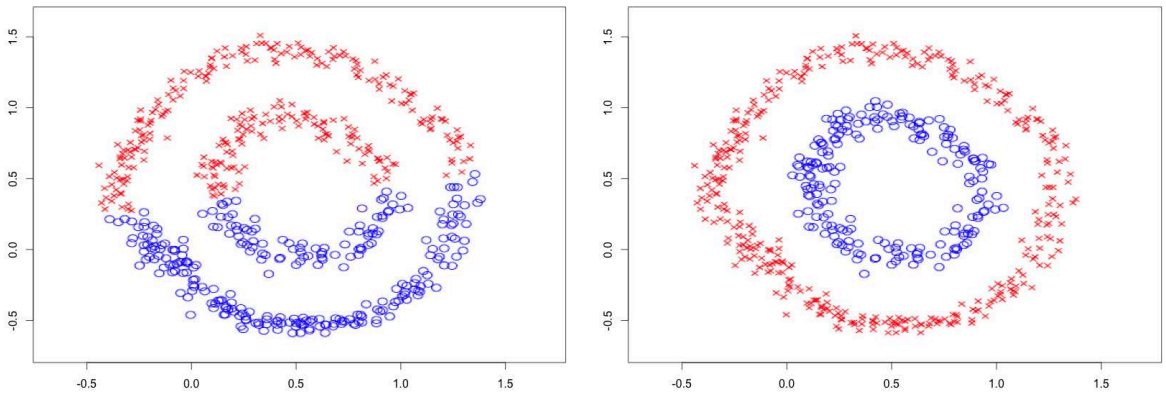


Figure 3.6: k-means clustering (left) vs. kernel k-means clustering (right)

The objective function that kernel k-means tries to minimize is the clustering error in feature space, and can be written as:

$$\underset{\{\mathbf{C}_1,...,\mathbf{C}_k\}}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \mathbf{C}_j} \|\Phi(\mathbf{x}_i) - \boldsymbol{m}_j\|^2 \tag{3.4}$$

while $(\mathbf{C}_1, ..., \mathbf{C}_k)$ are k clusters in feature space, Φ is a nonlinear transformation, and the centroid or pre-image of cluster $\mathbf{C}_j$ is denoted by $\boldsymbol{m}_j$, defined as:

$$\boldsymbol{m}_j = \frac{1}{|\mathbf{C}_j|} \sum_{\mathbf{x}_i \in \mathbf{C}_j} \Phi(\mathbf{x}_i) \tag{3.5}$$

Note that, cluster centroids in the feature space cannot be computed. We don't have explicit knowledge of representations of transformation function $\Phi(.)$ and the inverse of Φ to map from feature space back to input space typically does not exist (the pre-image problem). Therefore, on can expand the minimization problem to solve it. If we expand the distance computation $\|\Phi(\mathbf{x}_i) - \boldsymbol{m}_j\|^2$ in the objective function (Eq. 3.4) by using the scalar product and the kernel trick[2], we obtain the following:

$$\|\Phi(\mathbf{x}_i) - \boldsymbol{m}_j\|^2 = \left(\Phi(\mathbf{x}_i).\Phi(\mathbf{x}_i) \; - \; \frac{2 \sum_{\mathbf{x}_{i'} \in \mathbf{C}_j} \Phi(\mathbf{x}_i).\Phi(\mathbf{x}_{i'})}{|\mathbf{C}_j|} \; + \; \frac{\sum_{\mathbf{x}_{i'}, \mathbf{x}_{i''} \in \mathbf{C}_j} \Phi(\mathbf{x}_{i'}).\Phi(\mathbf{x}_{i''})}{|\mathbf{C}_j|^2} \right) \tag{3.6}$$

Hence only inner products are used in the computation of the Euclidean distance between a data point and a centroid of the cluster in the high-dimensional feature space. As a result, if we are given a kernel matrix K, where $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_{i'}) = \Phi(\mathbf{x}_i).\Phi(\mathbf{x}_{i'})$, the clustering can be performed without knowing explicit representations of $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_{i'})$.

[2] $\|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\|^2 = \kappa(\mathbf{x}_i, \mathbf{x}_i) - 2\kappa(\mathbf{x}_i, \mathbf{x}_j) + \kappa(\mathbf{x}_j, \mathbf{x}_j)$

While, kernel-based clustering algorithm can capture the nonlinear structure in data, they do not scale well in term of speed and memory requirements when the number of data points to be clustered exceeds of thousands. It requires computation and storage of $N \times N$ kernel matrix in memory, rendering it non-scalable to large dataset. But, what is the largest number of data points, N, that can be handled? Using an intel xeon e7-8837 processor machine, octa-core, 2.8GHz, 4TB max memory, with N greater than one million data points, may take several days to compute the kernel matrix alone. Thus, distributed and approximate versions of kernel k-means clustering algorithms recommended to handle large datasets.

Since a large fraction of attention from the data mining community has focuses on temporal data, of all the techniques applied clustering is perhaps the most frequently used, we are interested in clustering temporal data in this thesis.

3.2 Motivation

Clustering time series is an important task in providing useful information in various domain, and usually applied prior to any temporal data analysis or machine learning tasks, to extract groups of time series and highlight the main underlying dynamics. Recently, it has attracted huge concentration in the data mining community [BF98]; [WW00]; [KGP01]; [KLT03]. k-means-based clustering, viz. standard k-means, k-means++, and all its variations, is among the most popular clustering algorithms, because it provides a good trade-off between the quality of the solution obtained and its computational complexity [AV07]. However, the k-means clustering of temporal data under the widely used dynamic time warping (DTW) or the several well-established temporal kernels [BHB02]; [Cut11]; [Cut+07]; [KL83]; [Shi+02]; [MG14] is challenging, because estimating the cluster centroids requires aligning multiple temporal data simultaneously. Such alignments, referred to as multiple sequence alignments [CNH00]; [THG94], become computationally prohibitive, costly and impractical when the data size increases. To do so, some progressive and iterative heuristics have been proposed but are limited to the standard DTW and to the temporal data of the same global behavior [ACS03]; [CNH00]; [PKG11]; [THG94]. For temporal data clustering, to bypass the centroid estimation problem, the k-medoids and the kernel k-means [DGK04]; [Gir02] are generally used [Lia05]. For the k-medoids, a medoid is a good representative of data that has the same global dynamic (or behavior) but inappropriate for capturing local temporal features [FDCG13]. For the kernel k-means, although efficient for non-linearly separable clusters since centroids cannot be estimated in the Hilbert space, it refers to pairwise comparisons for the cluster assignment step. While k-means-based clustering, of linear complexity, remains a fast algorithm, the k-medoids and the kernel k-means clustering have a quadratic complexity due to the pairwise comparisons involved.

In this chapter, we propose a fast and accurate approach that generalizes the k-means based clustering algorithm for temporal data based on i) an extension of the standard time warp measures to consider both global and local temporal differences and ii) a tractable and fast estimation of the cluster representatives based on the extended time warp measures. Temporal

data centroid estimation is formalized as a non-convex quadratic constrained optimization problem. The developed solutions allow to estimate not only the temporal data centroid but also its weighting vector, which indicates the representativeness of the centroid elements. The solutions are particularly studied on four time warp measures that are commonly used in practice: the standard dynamic time warping [KL83], the dynamic temporal alignment kernel [Shi+02], the Gaussian dynamic time warping kernel [BHB02], and the global alignment kernel [Cut+07]; [Cut11].

3.3 Problem statement

The k-means algorithm aims at providing a partition of a set of data points in distinct clusters such that the inertia[3] within each cluster is minimized. The k-means clustering algorithm was originally developed with the Euclidean distance, the representative of each cluster being defined as the center of gravity of the cluster. This algorithm can, however, be generalized to arbitrary dissimilarities (resp. similarities) by replacing the representative update step with an explicit optimization problem that yields, for each cluster, the representative that minimizes (resp. maximizes) the total dissimilarity (resp. similarity) to all the data points of that cluster [SI84]; [Has+05]; [HH08]. Here we focus on (dis)similarity measures between time series that are based on time alignments because such measures (which encompass the standard dynamic time warping and its variants/extensions) are the ones most commonly used in the literature.

In the following, $\mathbf{X}$ denotes a set of univariate discrete time series $\mathbf{x} = (x_1, ..., x_T)$ of assumed length T[4]. An alignment $\boldsymbol{\pi}$ of length $|\boldsymbol{\pi}|$ between two time series is defined as the sequence of couples of aligned elements of the two time series. Intuitively, an alignment $\boldsymbol{\pi}$ between two time series describes a way to associate each element of one time series to one or more elements of another time series and vice-versa. Such an alignment can be simply represented by a path in the $T \times T$ grid, where the monotonicity conditions ensure that the alignment path is neither going back nor jumping. We will denote $\mathcal{A}$ as the set of all the alignments between two time series. For measures based on the time warp alignments, the integration of a weighting vector allows one to differently weigh the different time stamps of the series under consideration. This notion is used for the centroid (or representative) of each cluster $\mathbf{c} = (c_1, ..., c_T)$, which is no longer a simple data point (*i.e.* a simple time series) but rather a time series with an associated weighting vector $\mathbf{w} = (w_1, ..., w_T)$. The role of the weight vector is to indicate, for each cluster centroid (or representative), the importance of each time stamp, while this importance varying from cluster to cluster. Let d be a dissimilarity measure defined on $\mathbb{R}^T \times (\mathbb{R}^T)^2$, such that d provides a measure of dissimilarity between a given time series $\mathbf{x}$ and a weighted centroid $(\mathbf{c}, \mathbf{w})$. The generalized k-means clustering algorithm aims to find a partition of the data points in k clusters $(\mathbf{C}_1, ..., \mathbf{C}_k)$ such that the intra-cluster

[3] Inertia is defined as the sum of distances between any data point in the cluster and the centroid (or representative) of the cluster

[4] All the results can however be directly extended to multivariate time series, possibly of different lengths, as the temporal alignments, at the core of the (dis)similarity measures we consider, can be defined in those cases as well.

dissimilarity is minimized. The associated minimization problem for $\mathbf{x} \in \mathbf{X}$ can be written as:

$$\underset{\{\mathbf{C}_1,...,\mathbf{C}_k\}}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{\mathbf{x}\in\mathbf{C}_i} d(\mathbf{x}, (\mathbf{c}_i, \mathbf{w}_i)) \tag{3.7}$$

where $(\mathbf{c}_i, \mathbf{w}_i) \in \mathbb{R}^T \times \mathbb{R}^T$, the weighted representative of cluster $\mathbf{C}_i$, is defined by:

$$\begin{cases} (\mathbf{c}_i, \mathbf{w}_i) \;=\; \underset{\mathbf{c}\,,\,\mathbf{w}}{\operatorname{argmin}} \sum\limits_{\mathbf{x}\in\mathbf{C}_i} d(\mathbf{x}, (\mathbf{c}, \mathbf{w})) \\ \text{subject to} : \sum\limits_{t=1}^{T} w_t = 1, \quad w_t > 0, \;\forall t \end{cases} \tag{3.8}$$

where the given constraints guarantee that the problem is not degenerate. The generalized k-means clustering algorithm, given in Algorithm 2, is a direct generalization of the standard k-means algorithm in which the representative (or centroid) update step has been replaced with a more general representative update step that does not rely on the Euclidean distance.

Algorithm 2 Generalized k-means clustering $(\mathbf{X}, k)$

1: Input: $\mathbf{X}$, k
2: Output: $\{\mathbf{C}_1, ..., \mathbf{C}_k\}$
3: $p = 0$
4: Select k representatives $\{\mathbf{c}_1^{(p)}, ..., \mathbf{c}_k^{(p)}\}$, either randomly or through strategies *à la* k-means++ [AV07]
5: **repeat**
6: *Cluster assignment*
$\mathbf{C}_i^{(p)} \leftarrow \{\mathbf{x} \in \mathbf{X} \mid \mathbf{c}_i^{(p)} = \underset{\mathbf{c}_j, 1\le j\le k}{\operatorname{argmin}} d(\mathbf{x}, (\mathbf{c}_j^{(p)}, \mathbf{w}_j^{(p)}))\}, \;\; 1 \le i \le k$
7: *Representative update*
$(\mathbf{c}_i^{(p+1)}, \mathbf{w}_i^{(p+1)}) \leftarrow h(\mathbf{C}_i^{(p)})$
where the function $h : \mathcal{P}(\mathbf{X}) \to \mathbb{R}^T \times \mathbb{R}^T$ satisfies:

$$\sum_{\mathbf{x}\in\mathbf{C}_i^{(p)}} d(\mathbf{x}, h(\mathbf{C}_i^{(p)})) \le \sum_{\mathbf{x}\in\mathbf{C}_i^{(p)}} d(\mathbf{x}, (\mathbf{c}_i^{(p)}, \mathbf{w}_i^{(p)})) \tag{3.9}$$

8: $p \leftarrow p + 1$
9: **until** all clusters $\{\mathbf{C}_1, ..., \mathbf{C}_k\}$ are stable
10: Return $\{\mathbf{C}_1, ..., \mathbf{C}_k\}$

The function h (line 7 of Algorithm2), referred to as the *representative* function, provides, from a set of time series points, a point in $\mathbb{R}^T$ with an associated weighting vector ($h : \mathcal{P}(X) \to \mathbb{R}^T \times \mathbb{R}^T$). As soon as the function h satisfies the condition expressed in Inequality (3.9), the intra-cluster dissimilarity decreases in the representative update step. This decrease is maximum when the function h is defined using Eq. (3.8)[5]. The intra-cluster dissimilarity also decreases in the cluster assignment step, which is identical to the one used in the standard k-means clustering algorithm. Thus the generalized k-means clustering algorithm provided above converges [SI84]; [Has+05]; [HH08]. The above formulation can of course be adapted

[5]Depending on the dissimilarity measure used, it may not be possible to obtain the point that minimizes Eq.(3.8), and therefore "looser" functions, based on Inequality (3.9), have to be considered.

to the similarity measures, by substituting the dissimilarity d with a similarity s and $\arg\min$ with $\arg\max$ in Eqs. (3.7) and (3.8), and by considering representative estimation functions h satisfying:

$$\sum_{\mathbf{x}\in\mathbf{C}_i^{(p)}} s(\mathbf{x}, h(\mathbf{C}_i^{(p)})) \geq \sum_{\mathbf{x}\in\mathbf{C}_i^{(p)}} s(\mathbf{x}, (\mathbf{c}_i^{(p)}, \mathbf{w}_i^{(p)}))$$

This condition indicates that the solution obtained at iteration p, $h(\mathbf{C}_i^{(p)})$, should be better than the solution which obtained at the previous iteration, $(\mathbf{c}_i^{(p)}, \mathbf{w}_i^{(p)})$. In the following, we first give the definition of the extended time warp measures, and then under these metrics we formalize the problem of centroid estimation in the representative update step.

3.4 Extended time warp measures

We focus in this thesis on four time warp measures that are commonly used in practice: the Dynamic Time Warping (DTW) [KL83], which is a dissimilarity measure, the Dynamic Temporal Alignment Kernel (κ_{DTAK}) [Shi+02], the Gaussian Dynamic Time Warping (κ_{GDTW}) [BHB02], and the Global Alignment kernel (κ_{GA}) [Cut+07]; [Cut11], which are three similarity measures and constitute a reference in kernel machines in several domains, such as computer vision, speech recognition or machine learning [Nom02]; [Bai12]; [ZTH13]. To take into account both global and local temporal differences, we propose an extension of the four measures with a weighted centroid, as discussed above. The extensions mainly introduce a weighted warping function that guides the learned alignments according to the representative (or centroid) elements importance to capture both global and local temporal features [SKDCG16b].

3.4.1 Weighted dynamic time warping (WDTW)

Definition 3.1
The Weighted Dynamic Time Warping (WDTW) *between the time series* $\mathbf{x}$ *and the weighted time series* $(\mathbf{c}, \mathbf{w})$ *is defined by:*

$$\mathbf{WDTW}(\mathbf{x}, (\mathbf{c}, \mathbf{w})) = \min_{\boldsymbol{\pi}\in\mathcal{A}} \underbrace{\frac{1}{|\boldsymbol{\pi}|} \sum_{(t',t)\in\boldsymbol{\pi}} f(w_t)\, \varphi(x_{t'}, c_t)}_{C(\boldsymbol{\pi})} = C(\boldsymbol{\pi}^*) \tag{3.10}$$

where $\mathcal{A}$ *(as before) is the set of all alignments possible between two time series,* $f : (0, 1] \to \mathbb{R}^+$ *is a non-increasing function and* $\varphi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ *is a positive, real-valued, dissimilarity function.*

The cost function C computes the sum of the weighted dissimilarities φ between $\mathbf{x}$ and $(\mathbf{c}, \mathbf{w})$ through the alignment $\boldsymbol{\pi}$. When the weights are uniform (or f is a constant function) and when φ is the Euclidean distance, Eq. 3.10 corresponds to the well known *Dynamic Time Warping* (DTW) [KL83]; [SK83]. The above definition thus generalizes the DTW to the case where the different instants are weighted. The fact that f is non-increasing guarantees that the most important instants (*i.e.* the instants with the higher weights) of the centroid $\mathbf{c}$ should be privileged in the optimal alignment that minimizes the cost function C. Lastly, the optimal alignment $\boldsymbol{\pi}^*$ is obtained through the same dynamic programming procedure as the one used for the standard dynamic time warping.

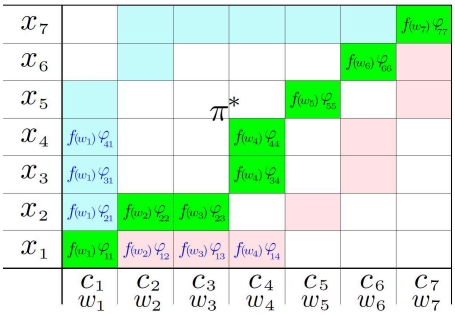


Figure 3.7: Three possible alignments are displayed between $\mathbf{x} = (x_1, ..., x_7)$ and $\mathbf{c} = (c_1, ..., c_7)$ and $\mathbf{w} = (w_1, ..., w_7)$ in the 7×7 grid. The value of each cell is the weighted divergence $f(w_t)\,\varphi_{t't} = f(w_t)\,\varphi(x_{it'}, c_t)$ between the aligned elements $x_{t'}$ and c_t. The optimal path $\boldsymbol{\pi}^*$ (the green one) that minimizes the average weighted divergence is given by $\pi_1 = (1, 2, 2, 3, 4, 5, 6, 7)$ and $\pi_2 = (1, 2, 3, 4, 4, 5, 6, 7)$.

The complexity of computing the extended time warp measure WDTW is $O(T^2)$, because the optimal alignment is obtained by dynamic programming[6]. However, one can speed it up by considering instead of $\mathcal{A}$, a subset of alignments usually around the diagonal [SC71]; [Ita75]; [SC78].

3.4.2 Weighted dynamic temporal alignment kernel ($\mathbf{WK}_{\mathrm{DTAK}}$)

More recently, several DTW-based temporal kernels that allow one to process time series with kernel machines have been introduced. The most common is the Dynamic Temporal Alignment Kernel (κ_{DTAK}) [Shi+02]. Here, we propose an extension of the pseudo-positive definite kernel κ_{DTAK}.

[6] T is length of each time series.

Definition 3.2
The Weighted Dynamic Temporal Alignment Kernel ($\mathrm{WK}_{\mathrm{DTAK}}$) *between the time series* $\mathbf{x}$ *and the weighted time series* $(\mathbf{c}, \mathbf{w})$ *is defined by:*

$$\mathbf{WK}_{\mathrm{DTAK}}(\mathbf{x}, (\mathbf{c}, \mathbf{w})) = \max_{\boldsymbol{\pi} \in \mathcal{A}} \underbrace{\frac{1}{|\boldsymbol{\pi}|} \sum_{(t', t) \in \boldsymbol{\pi}} f(w_t)\, \kappa(x_{t'}, c_t)}_{C_\kappa(\boldsymbol{\pi})} = C_\kappa(\boldsymbol{\pi}^*) \tag{3.11}$$

where $f : (0, 1] \to \mathbb{R}^+$ *is a non-decreasing function and* κ *is a Gaussian kernel* ($e^{-\frac{1}{2\sigma^2}\|x-y\|^2}$) *with an associated free parameter* σ *corresponding to the standard deviation which used to measure the similarity between aligned elements.*

When the weights are uniform (or the warping function f is a constant function) and when κ is the standard Gaussian kernel, the Eq. 3.11 corresponds to the Dynamic Time Alignment Kernel (κ_{DTAK}) introduced in [Shi+02]. The non-decreasing property of f ensures that the most important instants are privilaged in the optimal alignment that maximizes the cost function C_κ. The complexity of $\mathrm{WK}_{\mathrm{DTAK}}$, similar to the WDTW, is quadratic in T, since the optimal alignment is again obtained by dynamic programming.

3.4.3 Weighted Gaussian dynamic time warping ($\mathbf{WK}_{\mathrm{GDTW}}$)

Definition 3.3
The Weighted Gaussian Dynamic Time Warping Kernel ($\mathrm{WK}_{\mathrm{GDTW}}$) *between the time series* $\mathbf{x}$ *and the weighted time series* $(\mathbf{c}, \mathbf{w})$ *is defined by:*

$$\mathbf{WK}_{\mathrm{GDTW}}(\mathbf{x}, (\mathbf{c}, \mathbf{w})) = \max_{\boldsymbol{\pi} \in \mathcal{A}} \underbrace{exp\left[\frac{-1}{\lambda} \frac{1}{|\boldsymbol{\pi}|} \sum_{(t', t) \in \boldsymbol{\pi}} f(w_t)\ \ \varphi(x_{t'}, c_t)\right]}_{C_\kappa(\boldsymbol{\pi})} = C_\kappa(\boldsymbol{\pi}^*) \tag{3.12}$$

where $f : (0, 1] \to \mathbb{R}^+$ *is a non-increasing function and* $\varphi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ *is an Euclidean norm.*

The cost function C_k computes the exponential of the weighted dynamic time warping distance between $\mathbf{x}$ and $(\mathbf{c}, \mathbf{w})$ through the alignment $\boldsymbol{\pi}$. When the weights are uniform (or f is a constant function), Eq. 3.11 corresponds to the Gaussian Dynamic Time Warping kernel (κ_{GDTW}) introduced in [BHB02].

As before, the optimal alignment $\boldsymbol{\pi}^*$ is obtained through the same dynamic programming procedure as the one used for the standard dynamic time warping.

3.4.4 Weighted kernel global alignment ($\mathbf{WK}_{\mathrm{GA}}$)

Similarly, the extension of the Global Alignment kernel [Cut+07], which defines a true positive definite kernel, under fair conditions, on the basis of all of the alignments $\boldsymbol{\pi} \in \mathcal{A}$, can be defined as follows.

Definition 3.4
The Weighted Global Alignment Kernel ($\mathrm{WK}_{\mathrm{GA}}$) *between the time series* $\mathbf{x}$ *and the weighted time series* $(\mathbf{c}, \mathbf{w})$ *is defined by:*

$$\mathbf{WK}_{\mathrm{GA}}(\mathbf{x}, (\mathbf{c}, \mathbf{w})) = \sum_{\boldsymbol{\pi} \in \mathcal{A}} \underbrace{\prod_{(t',t) \in \boldsymbol{\pi}} f(w_t)\, \kappa(x_{t'}, c_t)}_{C_\kappa(\boldsymbol{\pi})} = C_\kappa(\boldsymbol{\pi}^*) \tag{3.13}$$

where $f : (0, 1] \to \mathbb{R}^+$ *is a non-decreasing function and* κ *is a local similarity kernel induced from the divergence* φ *as* $\kappa = e^{-\lambda\varphi}$.

The centroid $\mathbf{c}$, with its weight vector $\mathbf{w}$, of a set of time series $\mathbf{X}$ corresponds to the weighted time series that minimizes (or maximizes) a dissimilarity (or similarity) measure with all the elements of $\mathbf{X}$. Considering respectively WDTW, $\mathrm{WK}_{\mathrm{DTAK}}$, $\mathrm{WK}_{\mathrm{GDTW}}$ and $\mathrm{WK}_{\mathrm{GA}}$ metrics defined above as dissimilarity and similarity measures, we obtain following optimization problems, solutions of which correspond to the weighted centroids:

$$\begin{cases} \textsc{wdtw} \ \text{Problem} \\ \underset{\mathbf{c},\mathbf{w}}{\operatorname{argmin}}\, g(\mathbf{c}, \mathbf{w}) \ = \ \underset{\mathbf{c},\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{N} \textsc{wdtw}(\mathbf{x}_i, (\mathbf{c}, \mathbf{w})) \end{cases}$$

$$\begin{cases} \textsc{wk}_{\mathrm{DTAK}} \ \text{Problem} \\ \underset{\mathbf{c},\mathbf{w}}{\operatorname{argmax}}\, g_\kappa(\mathbf{c}, \mathbf{w}) = \underset{\mathbf{c},\mathbf{w}}{\operatorname{argmax}} \sum_{i=1}^{N} \textsc{wk}_{\mathrm{DTAK}}(\mathbf{x}_i, (\mathbf{c}, \mathbf{w})) \end{cases}$$

$$\begin{cases} \textsc{wk}_{\mathrm{GDTW}} \ \text{Problem} \\ \underset{\mathbf{c},\mathbf{w}}{\operatorname{argmax}}\, g_\kappa(\mathbf{c}, \mathbf{w}) = \underset{\mathbf{c},\mathbf{w}}{\operatorname{argmax}} \sum_{i=1}^{N} \textsc{wk}_{\mathrm{GDTW}}(\mathbf{x}_i, (\mathbf{c}, \mathbf{w})) \end{cases}$$

and

$$\begin{cases} \textsc{wk}_{\mathrm{GA}} \ \text{Problem} \\ \underset{\mathbf{c},\mathbf{w}}{\operatorname{argmax}}\, g_\kappa(\mathbf{c}, \mathbf{w}) \ = \ \underset{\mathbf{c},\mathbf{w}}{\operatorname{argmax}} \sum_{i=1}^{N} \textsc{wk}_{\mathrm{GA}}(\mathbf{x}_i, (\mathbf{c}, \mathbf{w})) \end{cases}$$

where $\boldsymbol{\pi}_i^*$ corresponds to the optimal alignment (*i.e.* the alignment of minimal cost for WDTW, or maximal cost for $\mathrm{WK}_{\mathrm{DTAK}}$, $\mathrm{WK}_{\mathrm{GDTW}}$ or $\mathrm{WK}_{\mathrm{GA}}$) between the time series $\mathbf{x}_i$ and the weighted time series $(\mathbf{c}, \mathbf{w})$, obtained through dynamic programming, subject to:

$$\textstyle\sum_{t=1}^{T} w_t = 1 \text{ and } w_t > 0, \ \forall t$$

Depending on the choice of f and φ, the restriction of g to $\mathbf{w}$ with $\mathbf{c}$ and $\mathbf{\Pi}^*$ fixed, denoted here $g(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*)$ with $\mathbf{\Pi}^* = (\boldsymbol{\pi}_1^*, ..., \boldsymbol{\pi}_N^*)$, as well as the restriction of g to $\mathbf{c}$ with $\mathbf{w}$ and $\mathbf{\Pi}^*$ fixed (denoted $g(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*)$), are not necessarily pseudo-convex[7]. This makes the WDTW problem difficult to solve and even ill-posed. The standard alternating procedure, that would amount here for the WDTW problem to alternately minimizing $g(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*)$ and $g(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*)$, is not guaranteed to converge and to find a local minimum. The same is true for the WK_{DTAK}, WK_{GDTW} or WK_{GA} problems, for which the pseudo-concavity of the functions $g_\kappa(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*)$ and $g_\kappa(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*)$ depends on the choice of f and κ. We establish here a theorem, the proof of which is given in Appendix A, that provides sufficient conditions on f and φ for the functions $g(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*)$ and $g(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*)$ to be pseudo-convex.

Theorem 3.1
Let $f : [0, 1] \to \mathbb{R}^+$ be the non-increasing function used in g, and let $\varphi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ be the positive, real-valued function used in g. Let us furthermore define:

$$\textbf{Condition 1:} \quad \forall a \in [-1, 1],\ \forall x \in [0, 1],\ \sum_{k=2}^{\infty} \frac{a^k}{k!} f^{(k)}(x) \geq 0$$
$$\textbf{Condition 2:} \quad \forall (c, c') \in \mathbb{R}^2, \forall x \in \mathbb{R}, \sum_{k=2}^{\infty} \frac{(c'-c)^k}{k!} \frac{\partial^k \varphi(x,c)}{\partial c^k} \geq 0$$

Then:
If Condition 1 holds, then $g(\boldsymbol{w}/\boldsymbol{c}, \mathbf{\Pi}^*)$ is pseudo-convex;
If Condition 2 holds, then $g(\boldsymbol{c}/\boldsymbol{w}, \mathbf{\Pi}^*)$ is pseudo-convex.

A similar theorem, also proven in Appendix A, can be established for f and κ, with conditions on their form so that the functions $g_\kappa(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*)$ and $g_\kappa(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*)$ are pseudo-concave.

Theorem 3.2
Let $f : [0, 1] \to \mathbb{R}^+$ be the non-decreasing function used in g_κ, let κ be the positive definite symmetric kernel used in g_κ and let Conditions 3 and 4 be defined as:

$$\textbf{Condition 3:} \quad \forall a \in [-1, 1],\ \forall x \in [0, 1],\ \sum_{k=2}^{\infty} \frac{a^k}{k!} f^{(k)}(x) \leq 0$$
$$\textbf{Condition 4:} \quad \forall (c, c') \in \mathbb{R}^2, \forall x \in \mathbb{R}, \sum_{k=2}^{\infty} \frac{(c'-c)^k}{k!} \frac{\partial^k \kappa(x,c)}{\partial c^k} \leq 0$$

Then:
If Condition 3 holds, then $g_\kappa(\boldsymbol{w}/\boldsymbol{c}, \mathbf{\Pi}^*)$ is pseudo-concave;
If Condition 4 holds, then $g_\kappa(\boldsymbol{c}/\boldsymbol{w}, \mathbf{\Pi}^*)$ is pseudo-concave.

These theorems allow one to choose functions that guarantee that the alternating approach to solve the WDTW, WK_{DTAK}, WK_{GDTW} or WK_{GA} problems is well-defined and converges towards local minima (for WDTW) and maxima (for WK_{DTAK}, WK_{GDTW} or WK_{GA}). We now proceed to the presentation of this alternating approach as well as of the functions we have retained.

[7]This is typically the case if φ is concave. A pseudo-convex function is a function that behaves like a convex function with respect to finding its local minima, but need not actually be convex.

3.5 Centroid estimation formalization and solution

We describe here the general strategy followed to estimate the representatives (or centroids) of a cluster of data points ($\mathbf{X}$), *prior* to study the solution. This strategy leads to the four extended time warp measures introduced above.

3.5.1 Representative update through alternative optimization

The problem given in Eq. 3.8 (and its maximization counterpart for similarity measures) can be solved by alternatively minimizing (or maximizing) for $\mathbf{c}$ and $\mathbf{w}$, with the other one being fixed, and by recomputing the optimal alignment warping path when necessary. This strategy is based on the following steps:

1. Set initial values for $\mathbf{c}$ and $\mathbf{w}$ and compute for these values $\mathbf{\Pi}^* = \{\boldsymbol{\pi}^*_{\mathbf{x}} \,/\, \mathbf{x} \in \mathbf{X}\}$, where $\mathbf{\Pi}^*$ denotes the optimal alignments warping path between $\mathbf{c}$ and the time series in $\mathbf{X}$.

2. Compute: $\underset{\mathbf{c}}{\operatorname{argmin}} \sum_{\mathbf{x}\in\mathbf{X}} d(\mathbf{x}, (\mathbf{c}, \mathbf{w}))$ (resp. $\underset{\mathbf{c}}{\operatorname{argmax}} \sum_{\mathbf{x}\in\mathbf{X}} s(\mathbf{x}, (\mathbf{c}, \mathbf{w}))$)

3. Compute: $\underset{\mathbf{w}}{\operatorname{argmin}} \sum_{\mathbf{x}\in\mathbf{X}} d(\mathbf{x}, (\mathbf{c}, \mathbf{w}))$ (resp. $\underset{\mathbf{w}}{\operatorname{argmax}} \sum_{\mathbf{x}\in\mathbf{X}} s(\mathbf{x}, (\mathbf{c}, \mathbf{w}))$),
   subject to: $\sum_{t=1}^{T} w_t = 1$ and $w_t > 0,\ \forall t$

4. Compute $\mathbf{\Pi}^*$ for the values of $\mathbf{c}$ and $\mathbf{w}$ obtained in the previous steps

5. Go back to step 2 until $\mathbf{c}$ and $\mathbf{w}$ are stable.

Note that for $\mathrm{WK}_{\mathrm{GA}}$, computing $\mathbf{\Pi}^*$ is not needed because this kernel is based on the sum of all alignments. Steps 2, 3 and 4 lead to a decrease (resp. increase) in the overall dissimilarity (resp. similarity) between the weighted centroid $(\mathbf{c}, \mathbf{w})$ considered at each iteration and the data points in $\mathbf{X}$. Indeed, each step respectively aims at finding values of $\mathbf{c}$, $\mathbf{w}$ and $\boldsymbol{\pi}$ that minimize (resp. maximize) the objective function, while the other quantities being fixed. Thus, the above strategy defines an algorithm that converges, and the associated function, which from $\mathbf{X}$ produces $(\mathbf{c}, \mathbf{w})$, is a valid representative function. The minimum (resp. maximum) in steps 2 and 3 can be obtained by computing the partial derivatives w.r.t $\mathbf{c}$ and $\mathbf{w}$ and either setting those partial derivatives to 0 and solving for $\mathbf{c}$ and $\mathbf{w}$ or using gradient descent approaches. Depending on the convexity (resp. concavity) properties of the measures, the minimum (resp. maximum) in step 2 and 3 may be local. This is not a problem *per se* because the procedure still converges while decreasing (resp. increasing) the inertia between the current pair $(\mathbf{c}, \mathbf{w})$ and the data points in $\mathbf{X}$. Now, we study the application of this strategy to each of the time warp measures retained by proposing the corresponding solutions.

3.5.2 Solution for WDTW

The extended form of the dynamic time warping measure is defined as:

$$\mathbf{WDTW}(\mathbf{x},(\mathbf{c},\mathbf{w})) = \min_{\boldsymbol{\pi}\in\mathcal{A}} \left(\frac{1}{|\boldsymbol{\pi}|} \sum_{(t',t)\in\boldsymbol{\pi}} f(w_t)\,\varphi(x_{t'},c_t) \right) \tag{3.14}$$

in which $f(w_t)$ is a non increasing warping function. The constraints, $\sum_{t=1}^{T} w_t = 1$ and $w_t > 0$, in the WDTW problem are based on quasi-concave functions (respectively an hyperplan and identity functions). Thus provided, that the functions f and φ in g satisfy Conditions 1 and 2 of Theorem 3.1, the Karush-Kuhn-Tucker conditions hold and the optimization problems in steps 2 and 3 can be solved with the method of Lagrange multipliers. At each step (2, 3 and 4), the function g is thus non-increasing and the overall procedure converges to a local minimum of g. For the function φ, we use here the standard Euclidean distance. In this case, Condition 2 of Theorem 3.1 amounts to

$$\sum_{k=2}^{\infty} \frac{(c'-c)^k}{k!} \frac{\partial^k \varphi(x,c)}{\partial c^k} = (c-c')^2 > 0$$

as all the partial derivatives of order greater than 2 are null. φ thus satisfies condition 2. For the function f, we consider here two different families:

$$f(x) = \begin{cases} x^{-\alpha} \\ e^{-\alpha x} \end{cases}$$

where $\alpha \in \mathbb{R}^+$ controls the influence of the weighting scheme. The negative exponent guarantees that the most important instants (*i.e.*, those highly weighted) are privileged in the optimal alignment that minimizes the WDTW. For $\alpha = 0$, Eq. 3.14 leads to the standard DTW.

The dominating term, for sufficiently smooth functions, in $\sum_{k=2}^{\infty} \frac{a^k}{k!} f^{(k)}(x)$ used in Condition 1 of Theorem 3.1 is obtained with the second order derivative (as $a \in [-1,1]$). Condition 1 thus amounts, for sufficiently smooth functions, to a convexity condition that is satisfied by the two smooth families of functions above. The solution of step 2 can thus be obtained by equating the partial derivative of $g(\mathbf{c}/\mathbf{w},\boldsymbol{\Pi}^*)$ with respect to $\mathbf{c}$ to 0 and solving for $\mathbf{c}$. The following proposition, the proof of which is given in Appendix B, provides the form of the solution obtained by doing so.

Given $\mathbf{w} = (w_1,...,w_T)$ and $\boldsymbol{\Pi}^* = \{\boldsymbol{\pi}^*_{\mathbf{x}} / \mathbf{x} \in \mathbf{X}\}$, the function defined in Eq. 3.14 is convex in $\mathbf{c}$ and the centroid (or representative) $\mathbf{c}$ that minimizes the sum of within cluster dissimilarities is obtained by solving the partial derivative equation, leading to, $\forall t$, $1 \le t \le T$:

$$c_t = \frac{\sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} x_{t'}}{\sum_{\mathbf{x}\in\mathbf{X}} \frac{|N(t,\mathbf{x})|}{|\boldsymbol{\pi}^*_{\mathbf{x}}|}} \tag{3.15}$$

where $\boldsymbol{\pi}^*_x$ denotes the optimal alignment path for $\mathbf{x} \in \mathbf{X}$ and $|N(t,\mathbf{x})| = \{t' \,/\, (t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}\}$ denotes the number of time instants of $\mathbf{x}$ aligned to time t of centroid $\mathbf{c}$.

For step 3, the solution for weight estimation of WDTW is obtained by equating the partial derivative of the Lagrangian of $g(\mathbf{w}/\mathbf{c}, \boldsymbol{\Pi}^*)$, subject to $\sum_{t=1}^{T} w_t = 1$ and $w_t > 0$, $\forall t$, with respect to $\mathbf{w}$ to 0 and solving for $\mathbf{w}$. The following propositions, also proven in Appendix B, provide the form of solutions obtained. Note that closed form solutions are obtained here for both step 2 and 3. Given $\mathbf{c} = (c_1, ..., c_T)$ and $\boldsymbol{\Pi}^* = \{\boldsymbol{\pi}^*_{\mathbf{x}} \,/\, \mathbf{x} \in \mathbf{X}\}$, the weight vector $\mathbf{w}$ that minimizes $g(\mathbf{w}/\mathbf{c}, \boldsymbol{\Pi}^*)$, $\forall t$, $1 \leq t \leq T$ is given by:

for $f(x) = x^{-\alpha}$;

$$w_t = \frac{A_t^{\frac{1}{1+\alpha}}}{\sum_{t=1}^{T} A_t^{\frac{1}{1+\alpha}}}$$

for $f(x) = e^{-\alpha x}$;

$$w_t = \frac{1}{\alpha} log \left(\frac{A_t}{(\prod_{t=1}^{T} A_t)^{1/T}} \right) + \frac{1}{T}$$

with

$$A_t = \sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} (x_{t'} - c_t)^2 \tag{3.16}$$

The solution to Eq. 3.8, corresponding to the representative update step of the generalized k-means clustering algorithm (Algorithm 2), is thus finally obtained through Algorithm 3.

Algorithm 3 *Centroid*$_{\text{WDTW}}$

1: Input: $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$
2: Output: ($\mathbf{c}$,$\mathbf{w}$)
3: Initialization:
- $\mathbf{c}^{(0)}$ randomly selected from $\mathbf{X}$
- $\mathbf{w}^{(0)} = (\frac{1}{T}, ..., \frac{1}{T})$
- $\mathbf{\Pi}^{*(0)}$: optimal alignments between $\mathbf{X}$ and $(\mathbf{c}^{(0)}, \mathbf{w}^{(0)})$
- $p = 0$

4: **repeat**
5: $p \leftarrow p + 1$
6: Update $c_t^{(p)}$ using Eq. 3.15, $1 \leq t \leq T$
7: Update $w_t^{(p)}$ using Eq. 3.16, $1 \leq t \leq T$
8: Update $\mathbf{\Pi}^{*(p)}$: optimal alignments between $\mathbf{X}$ and $(\mathbf{c}^{(p)}, \mathbf{w}^{(p)})$
9: **until** $(\mathbf{c}^{(p)}, \mathbf{w}^{(p)}) \approx (\mathbf{c}^{(p-1)}, \mathbf{w}^{(p-1)})$
10: Return ($\mathbf{c}$,$\mathbf{w}$)

3.5.3 Solution for $\mathbf{WK}_{\text{DTAK}}$

The extension of the κ_{DTAK} is defined as:

$$\mathbf{WK}_{\text{DTAK}}(\mathbf{x}, (\mathbf{c},\mathbf{w})) = \max_{\boldsymbol{\pi} \in \mathcal{A}} \left(\frac{1}{|\boldsymbol{\pi}|} \sum_{(t',t) \in \boldsymbol{\pi}} f(w_t)\, \kappa(x_{t'}, c_t) \right) \tag{3.17}$$

where a Gaussian kernel ($e^{-\frac{1}{2\sigma^2}\|x-y\|^2}$) with an associated free parameter σ corresponding to the standard deviation is used to measure the similarity between aligned elements. Similarly, the constraints, $\sum_{t=1}^{T} w_t = 1$ and $w_t > 0$, $\forall t$, provided that the functions f and κ in g_κ satisfy Conditions 3 and 4 of Theorem 3.2, the Karush-Kuhn-Tucker conditions hold and the optimization problems in steps 2 and 3 can be solved with the method of Lagrange multipliers. At each step (2, 3 and 4), the function g_κ is thus non-decreasing and the overall procedure converges to a local maximum of g_κ. We consider here a simple gaussian kernel for κ, that is at the basis of several DTW kernels, as in [Shi+02]; [Cut+07]; [ZTH08]; [Cut11]; [ZT12].

For the function f, we consider the two families of functions, which can be seen as the counterpart of the functions used for WDTW:

$$f(x) = \begin{cases} x^\alpha & \alpha < 1 \\ \log(\alpha x) & \alpha > 0 \end{cases}$$

In this case, f is a non-decreasing warping function that ensures that the most important instants are privileged in the optimal alignment. The standard κ_{DTAK} formulation is obtained with $\alpha = 0$. Note also that Eq. 3.17 can be viewed as a pre-image problem [HR11] that aims at estimating the pre-image, in the input space, of the barycenter of the data points in the feature space induced by the WK_{DTAK} kernel.

As before, the dominating term, for sufficiently smooth functions, in $\sum_{k=2}^{\infty} \frac{a^k}{k!} f^{(k)}(x)$ used in Condition 3 of Theorem 3.2 is obtained with the second order derivative (as $a \in [-1, 1]$).

Condition 3 thus amounts, for sufficiently smooth functions, to a concavity condition that is satisfied by the two smooth families of functions above.

Given $\mathbf{w} = (w_1, ..., w_T)$ and $\mathbf{\Pi}^* = \{\boldsymbol{\pi}^*_{\mathbf{x}} / \mathbf{x} \in \mathbf{X}\}$, the function defined in Eq. 3.17 is concave in $\mathbf{c}$. The solution of step 2 can be thus be obtained by equating the partial derivative of $g_\kappa(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*)$ with respect to $\mathbf{c}$ to 0 and solving to $\mathbf{c}$. We however know of no closed-form solution in this case, and resort to a stochastics gradient ascent method based on the following update rules at iteration p:

$$c_t^{(p+1)} = c_t^{(p)} + \eta^{(p)} \frac{\partial L}{\partial c_t^{(p)}}$$

and

$$\eta^{(p+1)} = \frac{\eta^{(p)}}{p} \ (\eta^{(0)} = 1)$$

with the partial derivative equation $\forall t$, $1 \leq t \leq T$:

$$\frac{\partial L}{\partial c_t} = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t) \frac{(x_{t'} - c_t)}{\sigma^2} \mathrm{e}^{(-\frac{(x_{t'} - c_t)^2}{2\sigma^2})} \tag{3.18}$$

where L is given by Eq. 3.17.

For the weights estimation, for step 3, the function defined in Eq. 3.17 is concave in $\mathbf{w}$ as $\alpha \in [0; 1]$. The solution of the partial derivative equation, obtained by equating the partial derivative of the Lagrangian, integrating the constraints on $\mathbf{w}$ to 0, can easily be obtained (proof is given in Appendix C). Given $\mathbf{c} = (c_1, ..., c_T)$ and $\mathbf{\Pi}^* = \{\boldsymbol{\pi}^*_{\mathbf{x}} / \mathbf{x} \in \mathbf{X}\}$, the weight vector $\mathbf{w}$ that maximizes the sum of intra-cluster similarities subject to $\sum_{t=1}^{T} w_t = 1$ and $w_t > 0, \ \forall t$, is defined by:

for $f(x) = x^\alpha$;

$$w_t = \frac{A_t^{\frac{1}{1-\alpha}}}{\sum_{t=1}^{T} A_t^{\frac{1}{1-\alpha}}}$$

for $f(x) = \log(\alpha x)$;

$$w_t = \frac{A_t}{\sum_{t=1}^{T} A_t}$$

with

$$A_t = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} e^{(-\frac{(x_{t'} - c_t)^2}{2\sigma^2})} \tag{3.19}$$

Algorithm 4 *Centroid*$_{\mathrm{WK_{DTAK}}}$

1: Input: $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$
2: Output: ($\mathbf{c}$,$\mathbf{w}$)
3: Initialization:
- $\mathbf{c}^{(0)}$ randomly selected from $\mathbf{X}$
- $\mathbf{w}^{(0)} = (\frac{1}{T}, ..., \frac{1}{T})$
- $\mathbf{\Pi}^{*(0)}$: optimal alignments between $\mathbf{X}$ and $(\mathbf{c}^{(0)}, \mathbf{w}^{(0)})$
- $p = 0$

4: **repeat**
5: $p \leftarrow p + 1$
6: // Update $c_t^{(p)}$, $1 \leq t \leq T$:
7: $q = 0,\ \eta = 1$
8: **repeat**
9: $q \leftarrow (q + 1),\ \eta \leftarrow \frac{\eta}{q}$
10: $c^{(q+1)} \leftarrow c^{(q)} + \eta \frac{\partial L}{\partial c}$, with $\frac{\partial L}{\partial c}$ given by Eq. 3.18
11: **until** $\frac{\partial L}{\partial c} \approx 0$
12: Update $w_t^{(p)}$ using Eq. 3.19, $1 \leq t \leq T$
13: Update $\mathbf{\Pi}^{*(p)}$: optimal alignments between $\mathbf{X}$ and $(\mathbf{c}^{(p)}, \mathbf{w}^{(p)})$
14: **until** $(\mathbf{c}^{(p)}, \mathbf{w}^{(p)}) \approx (\mathbf{c}^{(p-1)}, \mathbf{w}^{(p-1)})$
15: Return ($\mathbf{c}$,$\mathbf{w}$)

Algorithm 4 summarizes the steps required to solve the centroid estimation problem (Eq. 3.8) for $\mathrm{WK_{DTAK}}$, corresponding to the representative update step of the generalized k-means algorithm. The gradient ascent steps for estimating centroid $\mathbf{c}$ correspond to lines 4 to 11.

3.5.4 Solution for $\mathbf{WK}_{\mathrm{GDTW}}$

The extended form of the Gaussian dynamic time warping kernel is defined as:

$$\mathbf{WK}_{\mathrm{DTAK}}(\mathbf{x}, (\mathbf{c}, \mathbf{w})) = \max_{\boldsymbol{\pi} \in \mathcal{A}} \left(\underbrace{exp\left[\frac{-1}{\lambda} \frac{1}{|\boldsymbol{\pi}^*|} \sum_{(t',t) \in \boldsymbol{\pi}^*} f(w_t)\ (x_{t'} - c_t)^2 \right]}_{A(\mathbf{w},\mathbf{c})} \right) \tag{3.20}$$

Given $\mathbf{w} = (w_1, ..., w_T)$ and $\mathbf{\Pi}^* = \{\boldsymbol{\pi}^*_{\mathbf{x}} / \mathbf{x} \in \mathbf{X}\}$, considering $f(w_t)$ a non-increasing function, the solution can thus be obtained with the method of Lagrange multipliers. The solutions for both centroid and weight estimation lead however to no closed form solutions, and resort to a stochastic gradient ascent method based on the following update rules at iteration p, for the centroid $\mathbf{c}$:

$$c_t^{(p+1)} = c_t^{(p)} + \eta^{(p)}\ \frac{\partial L}{\partial c_t^{(p)}}$$

and

$$\eta^{(p+1)} = \frac{\eta^{(p)}}{p}(\eta^{(0)} = 1)$$

with

$$\frac{\partial L}{\partial c_t} = \frac{-2}{\lambda}\left[\sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|}\left(\sum_{t''\in T,(t'',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)(c_t - x_{t''})\right) . A(\mathbf{w},\mathbf{c})\right] \tag{3.21}$$

and for the weights vector $\mathbf{w}$:

$$w_t^{(p+1)} = w_t^{(p)} + \eta^{(p)}\ \frac{\partial L_w}{\partial c_t^{(p)}}$$

and

$$\eta^{(p+1)} = \frac{\eta^{(p)}}{p}(\eta^{(0)} = 1)$$

with

$$\frac{\partial L_w}{\partial w_t} = \frac{\alpha}{\lambda}\ w_t^{-(\alpha+1)}\ \left[\sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|}\ \left(\sum_{t''\in T,(t'',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} (x_{t''} - c_t)^2\right) . A(\mathbf{w},\mathbf{c})\right]$$

for $f(x) = x^{-\alpha}$, and (3.22)

$$\frac{\partial L_w}{\partial w_t} = \frac{\alpha}{\lambda}\ e^{(-\alpha\, w_t)}\ \left[\sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|}\ \left(\sum_{t''\in T,(t'',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} (x_{t''} - c_t)^2\right) . A(\mathbf{w},\mathbf{c})\right]$$

for $f(x) = e^{-\alpha x}$.

The solutions, for step 2 and 3, are obtained by equating the partial derivative of the Lagrangian (integrating the constraints) of $g_\kappa(\mathbf{c}/\mathbf{w}, \boldsymbol{\Pi}^*)$ with respect to $\mathbf{c}$ to 0 and solving for $\mathbf{c}$ and $g_\kappa(\mathbf{w}/\mathbf{c}, \boldsymbol{\Pi}^*)$ with respect to $\mathbf{w}$ to 0 and solving for $\mathbf{w}$. The proofs are given in Appendix D. Algorithm 5 summarizes the steps to solve the centroid estimation problem (Eq. 3.8) for WK_{GDTW}. Both centroid and weight estimated by a gradiant ascent method.

Algorithm 5 *Centroid*$_{\mathrm{WK_{GDTW}}}$

Input: $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$
Output: ($\mathbf{c}$,$\mathbf{w}$)
Initialization:
- $\mathbf{c}^{(0)}$ randomly selected from $\mathbf{X}$
- $\mathbf{w}^{(0)} = (\frac{1}{T}, ..., \frac{1}{T})$
- $\mathbf{\Pi}^{*(0)}$: optimal alignments between $\mathbf{X}$ and $(\mathbf{c}^{(0)}, \mathbf{w}^{(0)})$
- $p = 0$

repeat
  $p \leftarrow p + 1$
  // Centroid update
  $c_t^{(p)} \leftarrow$ Gradient-Ascent($c_t^{(p-1)}$, $w_t^{(p-1)}$, $\mathbf{\Pi}^{*(p-1)}$), $1 \leq t \leq T$
  // Weight update
  $w_t^{(p)} \leftarrow$ Gradient-Ascent($c_t^{(p)}$, $w_t^{(p-1)}$, $\mathbf{\Pi}^{*(p-1)}$), $1 \leq t \leq T$
  // Alignment update
  $\mathbf{\Pi}^{*(p)}$: optimal alignments between $\mathbf{X}$ and $(\mathbf{c}^{(p)}, \mathbf{w}^{(p)})$
until $g_\kappa(\mathbf{c}^{(p)}, \mathbf{w}^{(p)}) \approx g_\kappa(\mathbf{c}^{(p-1)}, \mathbf{w}^{(p-1)})$
Return ($\mathbf{c}$,$\mathbf{w}$)

3.5.5 The case of $\mathbf{WK}_{\mathrm{GA}}$

The extension of the κ_{GA}, for $f(w_t) = w_t^\alpha$, is defined as:

$$\mathbf{WK}_{\mathrm{GA}}(\mathbf{x}, (\mathbf{c},\mathbf{w})) = \sum_{\boldsymbol{\pi} \in \mathcal{A}} \prod_{(t',t) \in \boldsymbol{\pi}} w_t^\alpha \, e^{-\lambda \Phi(x_{t'}, c_t)} \tag{3.23}$$

with

$$\Phi(x_{t'}, c_t) = \frac{1}{2\sigma^2} \|x_{t'} - c_t\|^2 + log(2 - e^{-\frac{1}{2\sigma^2}\|x_{t'} - c_t\|^2})$$

where, similar to the $\mathrm{WK_{DTAK}}$, a Gaussian kernel is used as the similarity measure between aligned elements. α plays the same role as before and lies again in interval $[0; 1]$. The parameter λ is used to attenuate the diagonal dominance problem which arises mainly for time series of significantly different lengths. Although diagonal dominance leads to the positive definite Gram matrices, it reduces the performance of the measure in practice. Lastly, the complexity of $\mathrm{WK_{GA}}$ is similar to that of WDTW, $\mathrm{WK_{DTAK}}$ and $\mathrm{WK_{GDTW}}$ as one can rely on a recurrence formula to estimate the cost of a given pair (t', t) on the basis of the costs of the three previous pairs $(t'-1, t)$, $(t'-1, t-1)$ and $(t', t-1)$ [Cut+07]; [Cut11]. As for $\mathrm{WK_{DTAK}}$ and $\mathrm{WK_{GDTW}}$ the time computation can be reduced by considering only a subset of alignments; for $\mathrm{WK_{GA}}$, a fastest version exists based on a pairing of Gaussian and triangular kernels [Cut11].

Given $\mathbf{w} = (w_1, ..., w_T)$, the partial derivative of $\mathrm{WK}_{\mathrm{GA}}$ for updating the centroid $\mathbf{c}$ is, $\forall t$, $1 \leq t \leq T$:

$$\frac{\partial L}{\partial c_t} = \sum_{\mathbf{x} \in \mathbf{X}} \sum_{\boldsymbol{\pi} \in \mathcal{A}} \left(\sum_{(t',t) \in \boldsymbol{\pi}} -\frac{(x_{t'} - c_t)}{\sigma^2 \ (2 - e^{\frac{-1}{2\sigma^2}(x_{t'} - c_t)^2})} \right) \times \prod_{(t',t) \in \boldsymbol{\pi}} w_t^{\alpha} \ e^{-\lambda \Phi(x_{t'}, c_t)} \tag{3.24}$$

where $\Phi(x_{t'}, c_t)$ and L correspond to Eq. 3.23. We know that there is no closed-form solution to the equation $\frac{\partial L}{\partial c_t} = 0$. More importantly, the computation of the above derivative given in Eq. 3.24 cannot benefit from the recurrence formulas of $\mathrm{WK}_{\mathrm{GA}}$, which ensure an efficient computation of the measure (see proofs in Appendix E). Thus, if we want to use gradient ascend methods, we need to compute the scores associated with each alignment and sum them, which is impractical in situations where T is large (more than a few tens). For this reason, we will not use $\mathrm{WK}_{\mathrm{GA}}$ in generalized k-means.

In the remainder of this thesis, for the experiments, we thus focus on the following extended (dis)similarity measures for time series: 1) WDTW as defined by Eq. 3.14, 2) $\mathrm{WK}_{\mathrm{DTAK}}$ as defined by Eq. 3.17, and 3) $\mathrm{WK}_{\mathrm{GDTW}}$ as defined by Eq. 3.20.

3.6 Conclusion

The k-means-based clustering is among the most popular clustering algorithms, because it provides a good trade-off between the quality of solution obtained and its computational complexity. However, the k-means clustering of temporal data under the widely used dynamic time warping or temporal kernels is challenging, because estimating cluster centroids requires aligning multiple temporal data simultaneously. Such alignments, become computationally prohibitive, costly and impractical when the data size increases. For temporal data clustering, to bypass the centroid estimation problem, the k-medoids and the kernel k-means are generally used. But, while k-means-based clustering of linear complexity, remains a fast algorithm, the k-medoids and kernel k-means clustering have a quadratic complexity due to the pairwise comparisons involved.

Here, we introduced a generalized centroid-based clustering algorithm for temporal data under time warp measures. For this, we proposed an extension of the common time warp measures to capture both global and local differences, and a tractable, fast and efficient estimation of the cluster centroids, under the extended time warp measures. This cluster centroid estimation is formalized as a quadratic constrained optimization problem. Finally, the developed solutions allow for estimating not only the temporal data centroid but also its weight vector, which indicates the importance of centroid's elements. The solutions generalize the averaging problem to time series of distinct behaviors that share local characteristics, through several warping functions.

Chapter 4

Experimental study

Sommaire

In this chapter, we first describe the datasets retained to conduct our experiments to evaluate the effectiveness of the proposed approach, then specify the validation process, prior to present and discuss about the results obtained. Later, we mainly compare the proposed generalized k-means algorithm based on the extended time warp measures, to the alternative k-means clustering, standard k-medoids and kernel k-means clustering approaches. A wide range of datasets is used to evaluate the efficiency of the proposed approach, in the context of classification. The quantitative evaluation is finally completed by a visualization and qualitative comparison of the centroids obtained.

4.1 Data description

Our experiments are conducted on 24 public datasets[1]. The classes composing the datasets are known in advance and define the ground truth partitions. The experimentations are first carried out on standard well-known datasets, which define a favorable case for the averaging and clustering task as time series behave similarly within classes (e.g. CBF, CC and GUNPOINT), as illustrated in Figure 4.1. We then consider more complex datasets (e.g. BME, UMD and CONSSEASON). The second type of datasets is more challenging: a) it is related to time series that have different global behaviors within classes, while still sharing local characteristics, b) the commonly shared characteristics may appear at varying time stamps within the classes,

[1] UMD, BME at `http://ama.liglab.fr/~douzal/tools.html`, the rest of the data at `http://www.cs.ucr.edu/~eamonn/time_series_data/`

and c) be hidden by noise (see Figure 4.2). For example, BME includes two challenging classes BEGIN and END characterized by a small bell arising at the initial and final period respectively. The overall behavior may be different depending on whether the large bell is pointing upward or downward. UMD introduces more complexity with the classes UP and DOWN characterized by a small bell that may occur at different time stamps.

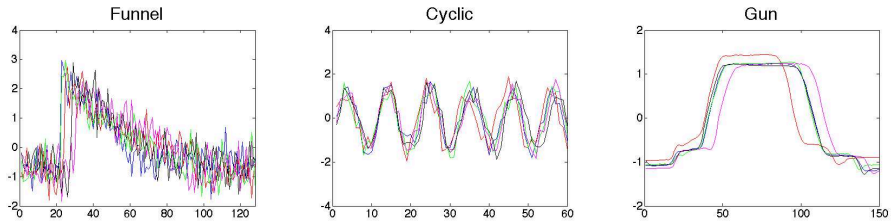


Figure 4.1: The time series behaviors within the classes "Funnel", "Cyclic" and "Gun" of the datasets CBF, CC and GUNPOINT, respectively.

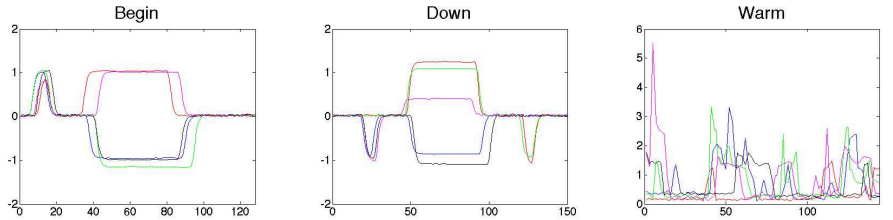


Figure 4.2: The time series behaviors within the classes "Begin", "Down" and "Warm" of the datasets BME, UMD and CONSSEASON, respectively.

To characterize these datasets, and because clustering methods such as k-means algorithms are known to be more efficient when the clusters are isotropic (*i.e.* spherical) and well isolated, for each dataset we computed the Bartlett's test of sphericity[2] [CL71]; [Jac93] as well as the isolation ratio. The isotropy is measured as the average of Bartlett's p-values for each cluster, as the p-value corresponds to the probability of being spherical. The cluster isolation ratio corresponds to the ratio of the sum of the dynamic time warping dissimilarities within clusters to the total DTW dissimilarities. The lower isolation ratio indicates the more isolated clusters. Table 4.1 describes the datasets considered, with their main characteristics: number of clusters (k), number of time series as dataset size (N), time series length (T), p-values for the isotropy and an isolation ratio. In particular, Bartlett's p-values ≤ 0.15 indicative of non-isotropic datasets, and *isolation ratio* ≥ 0.2, indicative of non-well-isolated datasets, are in bold. For instance, TWOPATTERNS is composed of clusters that are globally non-spherical but reasonably well-isolated, whereas YOGA clusters are relatively spherical but not well-isolated. As one can note from the Bartlett's p-values and isolation ratios displayed in Table 4.1, most of the datasets considered are composed of clusters that are either non-isotropic or not well-isolated (or both) and are thus challenging for the k-means and kernel k-means clustering algorithms [Gir02]. In addition, to visualize the underlying structure of the datasets, we performed a multidimensional scaling[3] [CC01] on the DTW pairwise dissimilarity matrix. Figure 4.3

[2] *Barspher* Matlab function
[3] *mdscale* Matlab function

displays the cluster structures on the first plan; with a stress[4] lower than 20%, while the representations obtained can be considered as accurate images of the underlying structures. Finally, as one can note, the datasets retained have very diverse structures (and shapes) and many of them are not linearly separable.

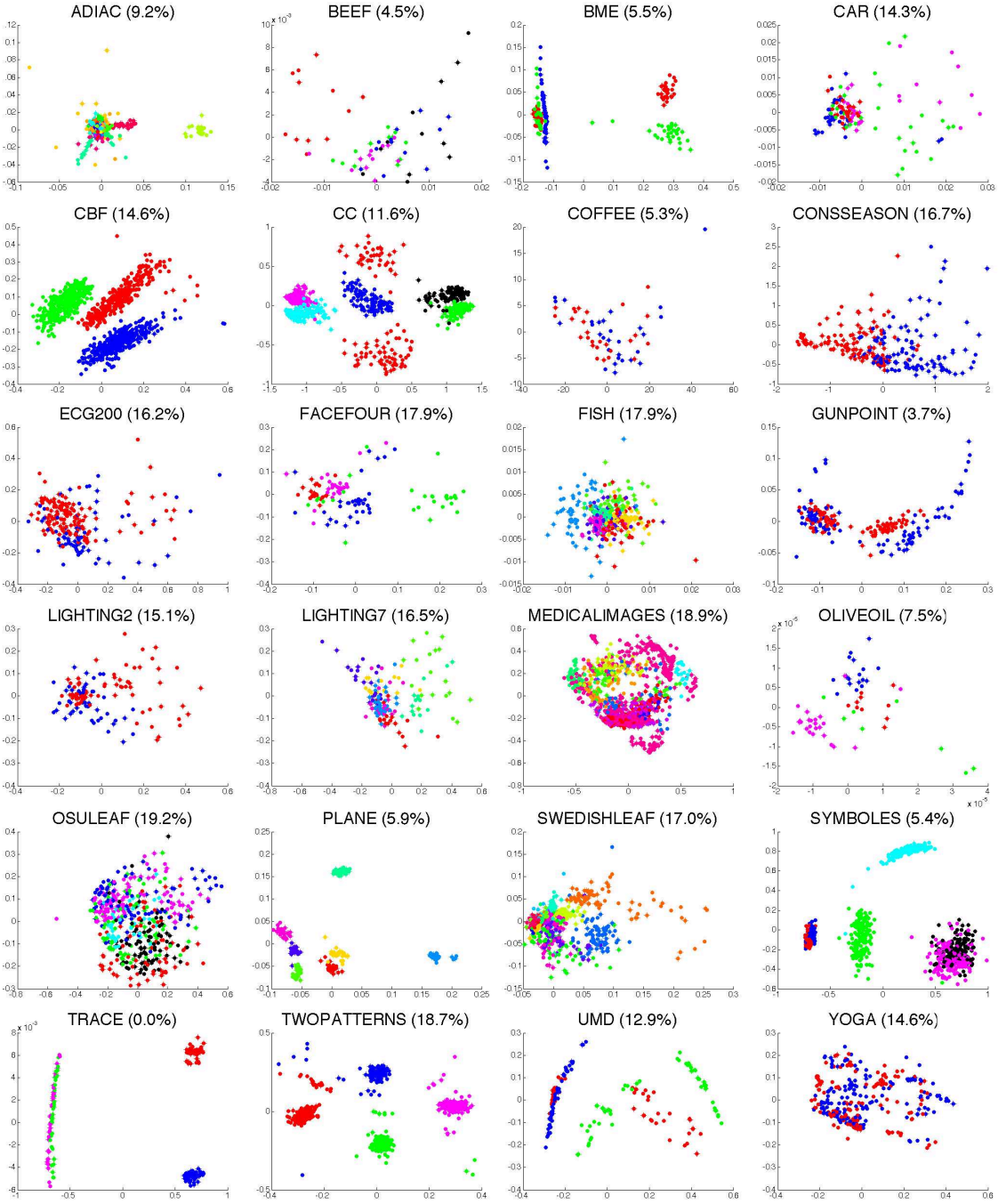


Figure 4.3: Structures underlying datasets

[4]Stress is defined as the error between the distances induced by the first plan and the original dissimilarities.

	Class Nb. k	Size(Train) N	Size(Test) N	TS. length T	Isotropy (p-value, std.)	Isolation *ratio*
ADIAC	37	390	391	176	(0.34, 0.31)	0.01
BEEF	5	30	30	470	(0.30, 0.35)	0.05
BME	3	30	150	128	**(0.00, 0.00)**	**0.25**
CAR	4	60	60	577	**(0.08, 0.06)**	**0.20**
CBF	3	30	900	128	**(0.00, 0.00)**	**0.23**
CC	6	300	300	60	**(0.06, 0.11)**	0.05
COFFEE	2	28	28	286	(0.46, 0.18)	**0.48**
CONSSEASON	2	180	180	144	**(0.19, 0.18)**	**0.48**
ECG200	2	100	100	96	(0.53, 0.27)	**0.50**
FACEFOUR	4	24	88	350	(0.39, 0.46)	0.19
FISH	7	175	175	463	(0.32, 0.33)	0.09
GUNPOINT	2	50	150	150	(0.24, 0.34)	**0.49**
LIGHTING2	2	60	61	637	(0.42, 0.59)	**0.52**
LIGHTING7	7	70	73	319	(0.23, 0.30)	0.08
MEDICALIMAGES	10	381	760	99	**(0.02, 0.03)**	**0.29**
OLIVEOIL	4	30	30	570	(0.29, 0.30)	0.15
OSULEAF	6	200	242	427	(0.29, 0.29)	0.16
PLANE	7	105	105	144	**(0.14, 0.24)**	0.01
SWEDISHLEAF	15	500	625	128	**(0.13, 0.28)**	0.03
SYMBOLES	6	25	995	398	(0.31, 0.36)	0.01
TRACE	4	100	100	275	**(0.15, 0.23)**	0.00
TWOPATTERNS	4	100	400	128	**(0.17, 0.26)**	0.10
UMD	3	36	144	150	**(0.00, 0.00)**	**0.20**
YOGA	2	30	300	426	(0.29, 0.04)	**0.50**

Table 4.1: Data description

4.2 Validation process

We compare here the proposed generalized k-means-based clustering algorithm (denoted as Gk-means) with the k-means based on the four well-known averaging methods (i.e., NLAAF, PSA, CWRT and DBA), k-medoids (k-med) under Euclidean and the standard DTW, and kernel k-means[5] (Kk-means) based on Euclidean, the standard κ_{GDTW}, κ_{DTAK}, κ_{GA} and κ_{TGA}. We focus here on the above approaches because they constitute the most frequently used variants of the k-means for time series clustering. As mentioned earlier, the k-medoids and the kernel k-means clustering algorithms are mainly used to bypass the centroid estimation process required by k-means with time warp measures. For our comparison, we rely on the Rand index[6] [Ran71], which measures the agreement between the obtained clusters and the ground truth ones, to evaluate each method. The Rand index lies in $[0, 1]$ and the higher index, the better the agreement is. In particular, the maximal value of Rand index, 1, is reached when the clusters (or partitions) are identical. For all the methods, the parameters taken within a validation set through a standard line/grid search process, as described in Table 4.2, and the parameters retained are the ones that maximize the Rand index on the validation set. Finally, the results reported hereafter are averaged after 10 repetitions of the corresponding algorithm. Because κ_{GDTW} and κ_{DTAK} are not strictly definite positive kernels [Cut+07]; [Cut11], we systematically added a scaled identity matrix δI to the Gram matrices with negative eigenvalues, while δ is the absolute value of the smallest negative eigenvalue and

[5] *kmedoids*, *kernelkmeans* Matlab functions
[6] *RandIndex* Matlab function

Method	Metric	Line / Grid values
Kernel	K_{GDTW}	$t \in \{0.2, 0.5, 1, 2, 5\} \odot \textbf{med}(\text{DTW}(\mathbf{x}, \mathbf{y}))$
k-means	K_{DTAK}	$\sigma \in \{0.2, 0.5, 1, 2, 5\} \odot \textbf{med}(\|\mathbf{x} - \mathbf{y}\|)$
	K_{GA}	$\sigma \in \{0.2, 0.5, 1, 2, 5\} \odot \textbf{med}(\|\mathbf{x} - \mathbf{y}\|)$, $\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$
	K_{TGA}	$\sigma \in \{0.2, 0.5, 1, 2, 5\} \odot \textbf{med}(\|\mathbf{x} - \mathbf{y}\|)$, $c \in \{0.25, 0.5\} \odot \textbf{med}(\|\mathbf{x}\|)$
Generelized	WDTW	$\alpha \in \{10^{-2}, 10^{-1}, 10^{0}\}$
k-means	WK_{GDTW}	$\alpha \in \{10^{-2}, 10^{-1}, 10^{0}\}$, $t \in \{0.2, 0.5, 1, 2, 5\} \odot \textbf{med}(\text{DTW}(\mathbf{x}, \mathbf{y}))$
	WK_{DTAK}	$\alpha \in \{10^{-2}, 10^{-1}, 10^{0}\}$, $\sigma \in \{0.2, 0.5, 1, 2, 5\} \odot \textbf{med}(\|\mathbf{x} - \mathbf{y}\|)$

Table 4.2: Parameter Line / Grid: The $\odot$ multiplication operator is element-wise (*e.g.* $\{1, 2, 3\} \odot \textbf{med} = \{\textbf{med}, 2\textbf{med}, 3\textbf{med}\}$). $\textbf{med}(\mathbf{x})$ stands for the empirical median of $\mathbf{x}$ evaluated on the validation set and $\mathbf{x}$ and $\mathbf{y}$ are vectors sampled randomly within time series in the validation set.

I is the identity matrix. For κ_{GA} method, the estimated value of the regularization parameter $\lambda \approx 1$ demonstrates that the problem of diagonal dominance has not been encountered for the retained datasets.

4.3 Experimental results

4.3.1 Results for time series clustering

In the context of clustering, the Rand indices and time consumptions for each method, on each dataset, are reported in Tables 4.3 and 4.4, respectively. Results in bold correspond to the best values, while the statistically significantly lower or higher ones (two-sided t-test at 5% risk),respectively, are shown with the down or up arrows.

Based on the Rand indices displayed in Table 4.3, one can note that Gk-means clustering method with WDTW leads to the best clustering results overall (23 datasets out of 24), followed by Gk-means method with WK_{DTAK}. Furthermore, the clustering approaches based on Gk-means introduced here outperform k-medoids and kernel k-means clustering almost on all datasets. Table 4.4 shows that Gk-means with WDTW leads almost to the fastest clustering method (18 datasets out of 24).

Impact of isotropy and isolation on clustering quality
Two most challenging datasets are apparently UMD and BME, with almost the lowest Rand indices varying in $[0.52, 0.62]$ for k-means, k-medoids and kernel k-means and in $[0.60, 0.69]$ for Gk-means clustering methods. In addition, UMD and BME are composed of weakly isolated and weakly spherical classes (see Table 4.1), include time series of distinct global behaviors within clusters [FDCG13]. Figure 4.4 shows the time series behaviors within the three classes BEGIN, MIDDLE and END of dataset BME, and Figure 4.5 illustrates the time series behaviors within the three classes UP, MIDDLE and DOWN of dataset UMD. Thus they require a similarity or dissimilarity measure able to capture the commonly shared features within clusters. Gk-means clustering method, by relying on weighted centroids, can capture

the characteristics shared locally within the clusters and improves the performances over the standard clustering methods, which fail to capture such local features.

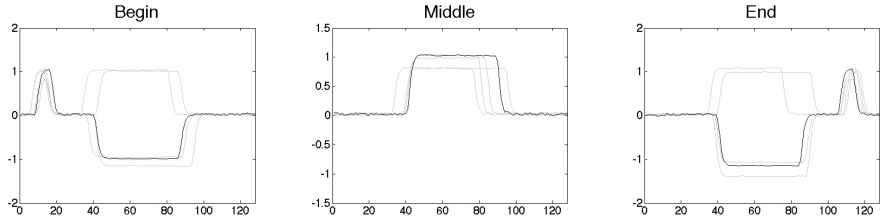


Figure 4.4: The time series behaviors within the three classes of dataset BME

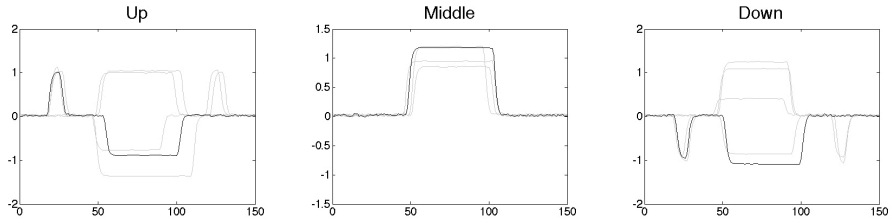


Figure 4.5: The time series behaviors within the three classes of dataset UMD

The second category of challenging datasets consists of COFFEE, ECG200 ,GUNPOINT, LIGHTING2 and YOGA and is composed of weakly isolated but highly (or mildly) spherical clusters (see Table 4.1). All the k-means clustering, k-medoids and kernel k-means methods lead to weak clustering results with a Rand index lying in $[0.48, 0.59]$, where the Gk-means clustering (with WDTW) leads to noticeably better clustering performances with a Rand index in $[0.60, 0.72]$. A third category of datasets is defined by CAR, CBF and MEDICALIMAGES is composed of mildly isolated but weakly spherical (or isotropic) clusters. Although k-means clustering, k-medoids and kernel k-means methods yield reasonable clustering performances, with a Rand index in $[0.50, 0.74]$, the Gk-means clustering (with WDTW) yields the best clustering results with a Rand index in $[0.77, 0.87]$. For the remaining datasets, composed of highly isolated and highly (or mildly) spherical clusters, good clustering results (Rand index in $[0.54, 0.97]$) are obtained using k-means clustering, k-medoids and kernel k-means methods, but are improved almost significantly by using Gk-means clustering (with WDTW) with a Rand index in $[0.75, 0.98]$. From the above results, we can obtain that all the k-means clustering, k-medoids and kernel k-means methods provide relatively good results even when the clusters are non-spherical (or non-isotropic); however, they are limited when the clusters are non-isolated. By contrast, Gk-means clustering with WDTW obtains very good clustering performances on all datasets, particularly on challenging datasets composed of clusters that are both non-spherical and not well-isolated.

	Kernel k-means (Kk-means)							k-medoid (k-med)			k-means				Generalized k-means (Gk-means)		
	RBF	k_{GDTW}	$k_{GDTW_{sc}}$	k_{DTAK}	$k_{DTAK_{sc}}$	k_{GA}	k_{TGA}	e_d	DTW	DTW_{sc}	NLAAF	PSA	CWRT	DBA	WDTW	wk_{DTAK}	wk_{GDTW}
ADIAC	0.749↓	0.916↓	0.922↓	0.915↓	0.932	0.876↓	0.920↓	0.943	0.952	0.950	0.867↓	0.909↓	0.911↓	0.939	**0.958**	0.950	0.944
BEEF	0.717	0.726	0.694	0.735	0.708	0.737	0.694	0.689	0.710	0.712	0.677	0.733	0.712	0.735	**0.752**	0.738	0.699
BME	0.600	0.597	0.603	0.584	0.599	0.615	0.581	0.602	0.609	0.603	0.524↓	0.551↓	0.563	0.574	**0.668**	0.605	0.626
CAR	0.699	0.672	0.694	0.565↓	0.577↓	0.493↓	0.558↓	0.688	0.584↓	0.649	0.635	0.694	0.567↓	0.650	**0.775**	0.712	0.701
CBF	0.675↓	0.688↓	0.677↓	0.712↓	0.706↓	0.739	0.702↓	0.666↓	0.712↓	0.706↓	0.652↓	0.708↓	0.692↓	0.740	**0.769**	0.755	0.709↓
CC	0.755↓	0.789↓	0.773↓	0.812↓	0.810↓	0.804↓	0.792↓	0.818↓	0.897	0.877↓	0.714↓	0.805↓	0.793↓	0.889↓	**0.935**	0.909	0.902
COFFEE	0.484	0.492	0.505	0.505	0.484	0.547	0.492	0.484	0.484	0.484	0.484	0.547	0.492	0.492	**0.605**	0.563	0.558
CONSSEASON	0.769	0.709	0.679↓	0.735	0.728	0.770	0.678↓	0.692↓	0.703	0.701↓	0.617↓	0.709	0.597↓	0.618↓	**0.792**	0.750	0.732
ECG200	0.564	0.560	0.575	0.578	0.583	0.590	0.576	0.544	0.519↓	0.584	0.505↓	0.524	0.507↓	0.534	**0.658**	0.608	0.553
FACEFOUR	0.719	0.742	0.714	0.744	0.707	0.697	0.539↓	0.721	0.714	0.619↓	0.590↓	0.613↓	0.629↓	0.677↓	**0.816**	0.774	0.749
FISH	0.707↓	0.800↓	0.825	0.828	0.802↓	0.844	0.822	0.624↓	0.768↓	0.766↓	0.708↓	0.779↓	0.766↓	0.807↓	**0.893**	0.839	0.827
GUNPOINT	0.503↓	0.495↓	0.498↓	0.502↓	0.499↓	0.497↓	0.503↓	0.548↓	0.497↓	0.497↓	0.491↓	0.497↓	0.497↓	0.517↓	0.719	**0.761**	0.529↓
LIGHTING2	0.511↓	0.531↓	0.499↓	0.503↓	0.508↓	0.496↓	0.492↓	0.536↓	0.563	0.495↓	0.495↓	0.522↓	0.522↓	0.531↓	**0.718**	0.540↓	0.533↓
LIGHTING7	0.793	0.801	0.820	0.827	0.816	0.815	0.818	0.802	0.816	0.816	0.788	0.806	0.794	0.839	**0.881**	0.835	0.829
MEDICALIMAGES	0.671↓	0.689↓	0.681↓	0.688↓	0.677↓	0.683↓	0.591↓	0.672↓	0.682↓	0.669↓	0.577↓	0.625↓	0.675↓	0.690↓	**0.866**	0.699↓	0.702↓
OLIVEOIL	0.825	0.827	0.806	0.832	0.797	0.826	0.814	0.747	0.793	0.797	0.703	0.819	0.723	0.767	**0.848**	0.833	0.820
OSULEAF	0.713↓	0.751↓	0.748↓	0.752↓	0.755↓	0.772↓	0.744↓	0.735↓	0.730↓	0.731↓	0.716↓	0.726↓	0.722↓	0.733↓	**0.914**	0.726↓	0.740↓
PLANE	0.923↓	0.886↓	0.829↓	0.914↓	0.921↓	0.909↓	0.842↓	0.948	0.977	0.974	0.908↓	0.899↓	0.896↓	0.967	**0.989**	0.950	0.939
SWEDISHLEAF	0.839↓	0.892	0.890	0.892	0.892	0.841↓	0.826↓	0.893	0.891	0.878	0.822↓	0.833↓	0.873↓	0.898	**0.909**	0.896	0.894
SYMBOLES	0.855↓	0.861↓	0.849↓	0.886↓	0.888↓	0.904	0.878↓	0.663↓	0.853↓	0.780↓	0.612↓	0.660↓	0.826↓	0.853↓	**0.927**	0.845↓	0.814↓
TRACE	0.753↓	0.747↓	0.713↓	0.822	0.799	0.751↓	0.744↓	0.751↓	0.840	0.832	0.695↓	0.717↓	0.814	0.873	**0.890**	0.876	0.796
TWOPATTERNS	0.628↓	0.792↓	0.656↓	0.696↓	0.623↓	0.684↓	0.615↓	0.627↓	0.842↓	0.681↓	0.666↓	0.733↓	0.787↓	0.841↓	**0.947**	0.852↓	0.799↓
UMD	0.614	0.610	0.552↓	0.623	0.552↓	0.621	0.609	0.600	0.606	0.586	0.533↓	0.562↓	0.546↓	0.574↓	**0.694**	0.652	0.628
YOGA	0.506↓	0.505↓	0.538	0.507↓	0.499↓	0.499↓	0.500↓	0.499↓	0.508↓	0.501↓	0.499↓	0.500↓	0.501↓	0.500↓	**0.603**	0.510↓	0.505↓

Table 4.3: Clustering Rand index

Relationships between methods and datasets

To compare globally the different clustering methods, we rely on a multiple correspondence analysis (MCA[7]), to analyze the 17 methods (considered as *individuals*) and the 24 datasets (considered as *categorical variables*). Multiple correspondence analysis is a data analysis technique for nominal categorical data and can be viewed as an extension of correspondence analysis (CA) which allows one to analyze the pattern of relationships of several categorical dependent variables. It can also incorporate quantitative variables. MCA is concerned with relationships within a set of variables, which usually are homogeneous, and allows the direct representation of individuals as points in geometric space. To do so, each method is described by a vector ("–", "+","++",...), with as many dimensions as there are datasets, in which the modalities "++", "+" and "–" indicate whether the Rand index of a method on a dataset is respectively highly greater, greater or lower than the mean Rand index obtained for that dataset over all the methods. Distinct groups of methods, corresponding to distinct ways to perform on the different datasets, can be distinguished. The first group is defined by the k-means using NLAAF, k-medoids and kernel k-means (with Euclidean) and is opposed to the other methods as it yields the lowest performances (corresponding to modality "–") particularly on CC, FISH and SYMBOLES. The second group is defined by the k-means using PSA and the three kernel k-means approaches (κ_{GA}, κ_{TGA} and $\kappa_{\text{GDTW}_{sc}}$) that yield the lowest performances on BEEF, BME, MEDICALIMAGES, OLIVEOIL, TRACE, SWEDISHLEAF and TWOPATTERNS.

To understand this limitation, we have studied the cluster structure induced by the dissimilarities based on the κ_{DTAK} and κ_{GA} Gram matrices. Figure 4.6 shows the projection obtained through MDS on the first plan for the dataset CC, with the dissimilarities[8] derived from κ_{DTAK}. As one can see, the red and blue clusters, non linearly separable with DTW (Figure 4.3 - second row and second column), are now well separated. The situation for κ_{GA} directly parallels the one for κ_{DTAK} .

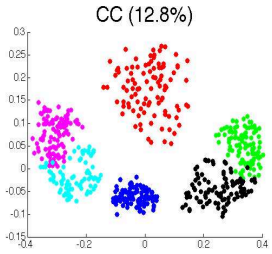


Figure 4.6: CC clusters based on κ_{DTAK} dissimilarities

Lastly, the last group includes the Gk-means approaches which yield the best performances (corresponding to modality "++") almost for all datasets. We can in particular observe that Gk-means with WDTW outperforms the other methods particularly on LIGHTING2, LIGHTING7, MEDICALIMAGES, OSULEAF, and YOGA.

The blue arrow in Figure 4.7 is known as the Guttman effect [LRR04]; it starts from

[7]We have used the MCA function available in the R package FactoMineR.

[8]The dissimilarities are defined as $d_{\text{DTAK}}(i,j) = \kappa_{\text{DTAK}}(i,i) + \kappa_{\text{DTAK}}(j,j) - 2\kappa_{\text{DTAK}}(i,j)$.

the upper left corner, that contains mainly datasets of modalities "−", then goes through the lower left corner, containing mixed modalities ("−" and "+"), through the lower right corner, again containing mixed modalities ("+" and "++") , and finally ends in the upper right corner containing datasets of modalities "++". The blue arrow suggests a global ranking of the methods based on their performance on all the datasets: Gk-means with WDTW is the best method, followed by Gk-means with $\mathrm{WK}_{\mathrm{DTAK}}$, followed by G$k$-means with $\mathrm{WK}_{\mathrm{GDTW}}$, k-means using DBA, κ_{DTAK} and k-medoids (with DTW), which yield similar results, and finally followed by the rest (see Figure 4.7).

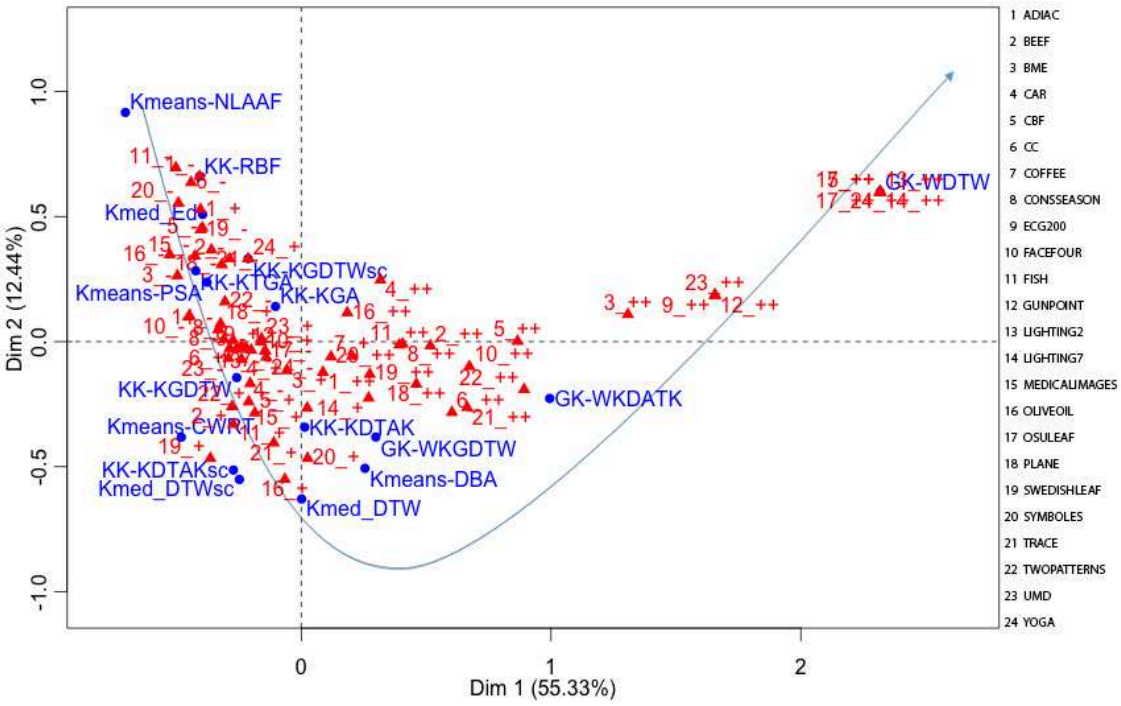


Figure 4.7: Global comparison of the clustering Rand index

Complexity and time considerations

Table 4.4 shows the time consumptions of the different clustering methods. As a reminder, results in bold correspond to the fastest methods with lowest time consumption, while the significantly higher ones are shown with the red up arrows. As one can note, Gk-means clustering is the fastest approach, with a approximately constant time requirement for all of the methods on small datasets (*e.g.* BEEF, COFFEE, OLIVEOIL). For large datasets (*e.g.* CBF, CC, MEDICALIMAGES, SWEDISHLEAF, SYMBOLES, YOGA), both k-medoids and kernel k-means clustering are significantly slower mainly due to the involved pairwise comparisons, whereas the Gk-means clustering, particularly with WDTW, remains remarkably fast. The standard k-means based on NLAAF and PSA averaging methods are highly time-consuming, largely because of the progressive increase of the centroid length during the pairwise combination process, which makes these approaches unusable for large time series datasets.

Figure 4.8 shows the multiple correspondence analysis of clustering time consumption to compare globally the clustering time consumption of different methods. The blue arrow starts

from the lower left corner, that contains mainly the fastest methods, then goes through the upper left corner and finally ends in the lower right corner containing the lowest clustering methods. The first group is defined by the k-means using CWRT, the standard Gaussian kernel k-means and k-medoids (with Euclidean), while all of these methods use Euclidean and therefore are not comparable with the other methods under time warp. Note that in clustering time consumption, Table 4.4, we ignored these three methods. The Gk-means approaches yield the fastest clustering results overall (18 datasets out of 24), followed by k-means with DBA. The third group is defined by kernel k-means (κ_{DTAK}, κ_{GDTW}, κ_{GA} and κ_{TGA}) and k-medoids (with DTW). Lastly, as mentioned, k-means with NLAAF and PSA with huge difference yield the lowest performances.

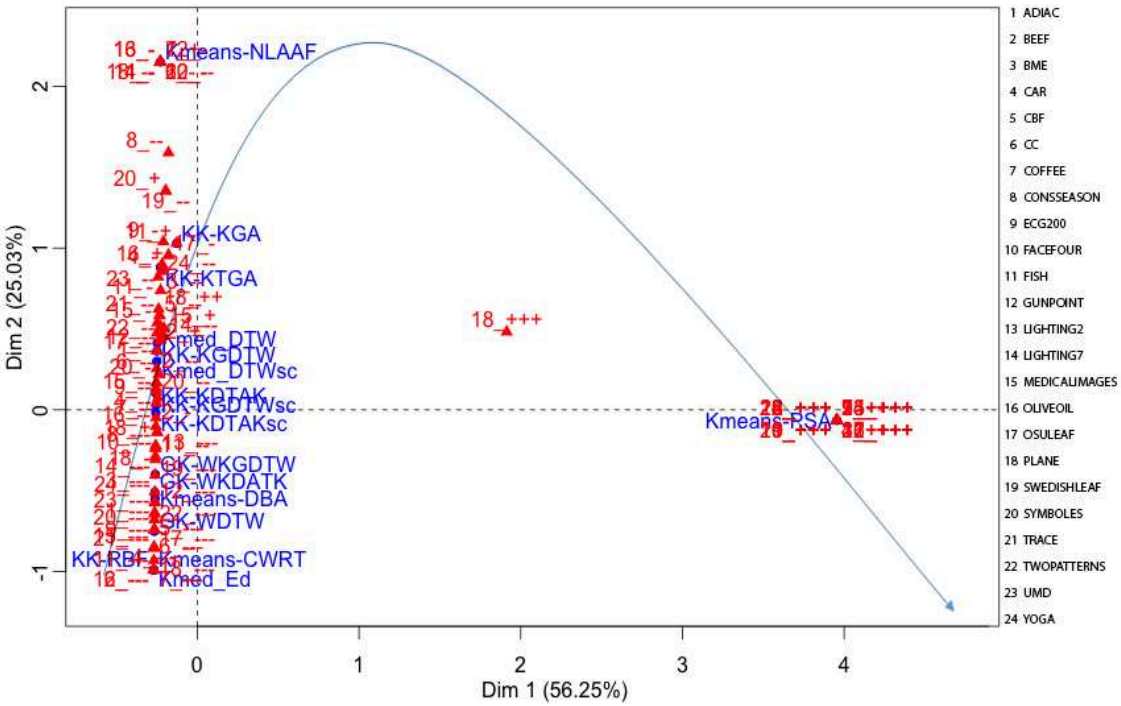


Figure 4.8: Global comparison of the clustering Time consumption

The obtained empirical results are in agreement with the theoretical complexities of each method. Let N be the number of time series to be clustered in k clusters, with the time series length T, and p the complexity of the measure between aligned instants ($p = 5$ for WDTW, $p = 8$ for WK_{DTAK}, $p = 9$ for WK_{GDTW}, and $p = 18$ for WK_{GA} - see Eqs. 3.14, 3.17, 3.20 and 3.23). Therefore, the complexity of Gk-means clustering is $O(pNkT^2)$, the one of k-medoids is $O(pN(N-k)T^2)$ and the complexity of kernel k-means clustering is $O(pN^2T^2)$, showing that kernel k-means clustering is slower than k-medoids, which is in turn slower than the Gk-means clustering. Lastly, the difference between Gk-means clustering with WDTW and WK_{DTAK} (or WK_{GDTW}) is mainly due to the ascent gradient part, while the difference between kernel k-means clustering with κ_{DTAK} and κ_{GA} is explained by the different values of p for each method.

	Kernel k-means (Kk-means)							k-medoid (k-med)			k-means				Generalized k-means (Gk-means)		
	RBF	k_{GDTW}	$k_{GDTW_{sc}}$	k_{DTAK}	$k_{DTAK_{sc}}$	k_{GA}	k_{TGA}	E_d	DTW	DTW_{sc}	NLAAF	PSA	CWRT	DBA	WDTW	wk_{DTAK}	wk_{GDTW}
ADIAC	0.9	714.8↑	604.3↑	722.1↑	676.0↑	992.2↑	875.6↑	1.1	899.6↑	742.5↑	1998.4↑	4003.4↑	95.4	399.1	**348.8**	374.4	400.3
BEEF	1.6	59.1	81.3	53.8	53.7	77.4	69.4	1.4	**49.8**	50.2	84.9	146.8↑	12.7	65.4	58.7	76.3	79.4
BME	2.9	79.8↑	66.8↑	63.2↑	60.6↑	130.5↑	105.3↑	2.7	102.5↑	93.2↑	131.8↑	464.2↑	4.9	21.7	**20.5**	45.6↑	59.8↑
CAR	0.8	285.2↑	247.8↑	314.4↑	270.0↑	491.5↑	460.1↑	1.2	336.3↑	276.7↑	591.4↑	1261.2↑	84.9	550.3↑	**167.7**	202.3	230.8
CBF	4.0	2482.9↑	2473.4↑	2912.2↑	2584.3↑	2874.8↑	2765.5↑	3.7	2414.9↑	2307.2↑	2499.4↑	9197.6↑	26.9	422.9↑	**361.9**	602.4↑	697.9↑
CC	1.7	112.6↑	122.8↑	122.4↑	119.8↑	162.8↑	133.7↑	0.9	122.9↑	125.4↑	177.7↑	488.7↑	2.9	64.9↑	**49.2**	139.7↑	152.3↑
COFFEE	1.1	23.0	23.7	18.8	16.2	17.1	20.2	0.9	16.8	16.9	39.2↑	75.9↑	2.6	22.5	**15.7**	23.3	24.1
CONSSEASON	2.3	155.4↑	140.2↑	164.9↑	139.8↑	297.6↑	261.3↑	1.1	173.2↑	160.3↑	504.3↑	1700.3↑	6.4	46.4↑	**35.8**	82.2↑	99.9↑
ECG200	1.5	15.8↑	28.6↑	12.6↑	13.7↑	26.4↑	25.9↑	1.2	15.3↑	14.9↑	33.4	71.2	1.9	**8.7**	10.2	12.2↑	14.7↑
FACEFOUR	1.9	335.8↑	329.5↑	363.8↑	252.7	343.4↑	275.9	1.8	**206.7**	213.8	609.7↑	2221.6↑	23.2	347.4↑	292.6↑	320.5↑	341.2↑
FISH	3.4	1493.2↑	1394.4↑	1103.5↑	1063.6↑	2520.8↑	2125.2↑	2.3	1525.5↑	1498.6↑	2010.8↑	4385.1↑	94.2	758.3	**663.1**	1007.1↑	1314.5↑
GUNPOINT	2.2	88.1↑	142.5↑	52.1↑	63.5↑	150.5↑	144.5↑	1.6	85.6↑	84.7↑	177.6↑	468.3↑	3.7	44.3	**39.4**	45.5	55.3↑
LIGHTING2	2.4	410.3↑	401.9↑	303.8↑	296.1↑	613.8↑	534.4↑	1.9	397.1↑	392.2↑	2977.7↑	6309.8↑	29.4	173.4	**141.9**	158.5	197.8
LIGHTING7	2.1	117.2↑	120.8↑	94.6	**82.4**	191.4↑	166.0↑	1.6	112.3	110.9	293.4↑	825.4↑	16.7	142.7↑	98.4	109.5	133.4↑
MEDICALIMAGES	7.3	968.6↑	988.4↑	943.6↑	706.2↑	1581.4↑	1616.3↑	2.4	940.2↑	927.7↑	1304.6↑	2897.5↑	21.9	321.3↑	**236.2**	1228.7↑	1538.2↑
OLIVEOIL	1.9	74.7	66.6	54.4	53.9	116.9↑	115.2↑	1.5	67.2	65.8	100.9↑	228.3↑	12.7	65.9	**52.2**	73.8	53.3
OSULEAF	3.6	2414.6↑	1717.7↑	1784.7↑	1735.2↑	3979.6↑	3456.9↑	2.5	1845.8↑	1804.7↑	2521.2↑	9989.8↑	94.2	915.6	**787.6**	921.8	1229.4↑
PLANE	1.1	37.9↑	33.3↑	33.6↑	30.6↑	64.6↑	49.5↑	0.9	45.5↑	38.4↑	**21.2**	66.6↑	8.6	32.4↑	30.0↑	37.9↑	40.1↑
SWEDISHLEAF	6.2	1139.2↑	1080.8↑	753.6↑	811.5↑	1741.3↑	1763.7↑	2.2	1083.5↑	1018.4↑	2003.3↑	7034.4↑	33.8	475.3↑	**350.8**	659.9↑	590.4↑
SYMBOLES	4.6	36211.2↑	34666.1↑	32623.3↑	28013.1↑	57909.9↑	53445.2↑	4.2	29320.2↑	28745.1↑	59874.3↑	99999.9↑	487.5	6014.2↑	**4121.1**	5711.8↑	6151.7↑
TRACE	1.2	258.9↑	244.8↑	250.6↑	232.4↑	310.9↑	351.7↑	1.1	233.8↑	240.5↑	397.2↑	1298.7↑	9.8	**78.9**	80.2	180.4↑	154.7↑
TWOPATTERNS	2.5	439.5↑	513.8↑	320.9↑	399.9↑	617.3↑	586.7↑	1.8	452.5↑	462.3↑	812.2↑	2143.4↑	12.8	167.8↑	**110.2**	172.9↑	288.8↑
UMD	2.2	93.3↑	82.7↑	91.4↑	81.4↑	165.5↑	137.2↑	2.1	105.7↑	99.2↑	124.4↑	503.9↑	5.5	33.5	**29.3**	55.5↑	74.3↑
YOGA	2.8	3227.4↑	3031.6↑	3133.2↑	3028.5↑	4988.7↑	4221.3↑	3.2	2030.2↑	2958.6↑	5536.8↑	18694.3↑	89.7	352.3↑	**280.2**	328.1	388.8↑

Table 4.4: Clustering time consumption(sec.)

4.3.2 Results for time series classification

In this section, we evaluate the relevance of the proposed estimated centroids and weights in a classification context. Among many algorithms to classify temporal data, the k-nearest neighbor with the DTW is performant. However, it is very time consuming as it needs to compare time series query with all training samples. While the idea of data reduction to mitigate this problem has a long history, classification based on nearest centroid could be one solution. To do so, we propose to use k-nearest weighted centroid classifier based on the above extended time warp measures (see Algorithm 6). Using generalized k-means-based clustering within each class, one can obtain a small number of weighted centroids per class.

Algorithm 6 Nearest weighted centroid classification ($\mathbf{X}_{train}$,$\mathbf{Y}_{train}$,$\mathbf{X}_{test}$,k)

Input: $\mathbf{X}_{train} = \{\mathbf{x}_1, ..., \mathbf{x}_{N_{train}}\}$
Input: $\mathbf{Y}_{train}$ = labels
Input: $\mathbf{X}_{test} = \{\mathbf{x}_1, ..., \mathbf{x}_{N_{test}}\}$
Input: k = class number
Output: class labels for $\forall \mathbf{x} \in \mathbf{X}_{test}$
for i : 1 to k **do**
    $(\mathbf{c}_i, \mathbf{w}_i)$ = **Centroid Estimation**($\mathbf{X}_{train}|\mathbf{Y}_{train} == i$) using Algo. 3, 4 or 5
end for
for $\forall \mathbf{x}_j \in \mathbf{X}_{test}$ **do**
    Classify $\mathbf{x}_j$ by nearest weighted centroid using Eq. 3.10, 3.11 or 3.12
end for
Return class labels for $\forall \mathbf{x} \in \mathbf{X}_{test}$

For the experiment, we here rely on the classification time consumption, presented in Table 4.7, while the main contribution is to speed up temporal classification. To verify the accuracy of the method, we have confidence in the classification error rate (see Table 4.5 and 4.6). The lower the error rate, the better the agreement is. Lastly, we consider a goodness criteria which is the ratio of "*fastness*" to "*accuracy*". For this, we define "*fastness*" as the normalized time consumption (so that all values are in (0,1)), and (1 - error rate) is the "*accuracy*". The lower value is the better classifier. Similar to the previous tables in clustering section, results in bold correspond to the best values, while the statistically significantly lower or higher ones (t-test at 5% risk) are represented by the down or up arrows, respectively.

Classification error rate

Based on the classification error rate displayed in Table 4.5, for k = 1, nearest neighbor classification methods lead to the best results in overall (23 datasets out of 24), followed by the proposed weighted centroid estimation methods. Using a small number of weighted centroids per class obtained by generalized k-means based clustering within each class, we can beat the nearest neighbor classifier. For example, for dataset BEEF, if we assume k=2 and we consider two weighted centroids per class, by generalized k-means based clustering, we outperform all the nearest neighbors classifiers by error rate = 0.433. A second example, for dataset OSULEAF, if we assume k=2, generalized k-means based clustering in each class,

obtains two weighted centroids per class. Thus, the classification error rate using WDTW will be 0.295, which outperforms all the other classifiers.

The experiments show that the proposed nearest weighted centroid classification methods significantly outperform the alternative nearest centroid classification approaches. To validate the claim, Table 4.6 presents all the nearest centroid classification error rate results. As one can note, classification using WDTW yields the best classification error rate overall (19 datasets out of 24), followed by WK_{DTAK}.

To compare globally the different nearest centroid classification methods, as before, we rely on MCA[9]. To do so, each method is described by a vector ("−", "+","++"), in which the modalities "++", "+" and "−" indicate whether the classification error rate of a method on a dataset is respectively highly greater, greater or lower than the mean error rate obtained for that dataset over all the methods. Lower error rate indicated better classification. According to the Figure 4.9, nearest centroid classification using PSA and NLAAF (to estimate the centroid) have the higher error rates almost for all the datasets, as they yield the lowest classification performances (corresponding to modality "++"). The group includes the the proposed nearest weighted centroid classification approaches which yield the best performances (corresponding to modality "-") almost for all datasets (23 datasets out of 24), followed by nearest neighbor classification using DBA. We can in particular observe that nearest weighted centroid classifier based on WDTW outperforms the other methods particularly on BME, CONSSEASON, FISH, SYMBOLES, TRACE, and YOGA.

The blue arrow suggests a global ranking of the methods based on their performance on all the datasets: it starts from upper left corner, that contains mainly datasets of modalities "-", then goes through the lower left corner, containing mixed modalities ("-" and "+"), through the lower right corner, containing mixed modalities ("+" and "++") , and finally ends in the upper right corner, again containing mixed modalities ("+" and "++"). The projection of the methods on the blue arrow suggest a global ranking: classification with nearest weighted centroid using WDTW is the best method, followed by WK_{DTAK}, DBA and WK_{GDTW} and finally followed by the rest.

[9]MCA function in the R package FactoMineR.

	k-NN kernel							k-NN (standard)			k-nearest centroid NLAAF	k-nearest centroid PSA	k-nearest centroid CWRT	k-nearest centroid DBA	k-nearest weighted centroid		
	RBF	k_{GDTW}	$k_{GDTW_{sc}}$	k_{DTAK}	$k_{DTAK_{sc}}$	k_{GA}	k_{TGA}	E_d	DTW	DTW_{sc}	DTW	DTW	DTW	DTW	WDTW	wk_{DTAK}	wk_{GDTW}
ADIAC	0.494↑	0.404	0.404	0.425	0.425	0.407	0.396	**0.389**	0.404	0.404	0.567↑	0.527↑	0.519↑	0.517↑	0.465↑	0.471↑	0.496↑
BEEF	**0.467**	0.500	0.500	0.500	**0.467**	0.500	**0.467**	**0.467**	0.500	0.500	0.600	0.533	0.600	0.567	0.533	0.600	0.600
BME	0.173↑	0.140↑	**0.007**	0.127↑	**0.007**	0.200↑	0.027	0.173↑	0.140↑	**0.007**	0.360↑	0.360↑	0.360↑	0.360↑	0.333↑	0.353↑	0.360↑
CAR	0.317	0.283	0.267	0.267	0.233	**0.200**	0.267	0.267	0.283	0.233	0.583↑	0.467↑	0.400↑	0.367↑	0.337	0.383↑	0.400↑
CBF	0.097↑	**0.011**	**0.011**	0.013	0.012	0.183↑	0.152↑	0.148↑	**0.011**	**0.011**	0.106↑	0.220↑	0.036↑	0.036↑	0.031↑	0.038↑	0.033↑
CC	0.120↑	0.023	**0.017**	0.023	0.033	0.177↑	0.083↑	0.120↑	0.023	**0.017**	0.113↑	0.377↑	0.037	0.027	0.027	0.087↑	0.110↑
COFFEE	0.250	**0.179**	**0.179**	**0.179**	**0.179**	**0.179**	0.214	0.250	**0.179**	**0.179**	0.464↑	0.464↑	0.464↑	0.464↑	0.464↑	0.464↑	0.464↑
CONSSEASON	**0.022**	0.111↑	0.072↑	0.089↑	0.056	**0.022**	**0.022**	**0.022**	0.111↑	0.072↑	0.350↑	0.389↑	0.172↑	0.133↑	**0.022**	0.244↑	0.311↑
ECG200	0.120	0.210↑	0.190	0.220↑	0.160	0.110	**0.100**	0.120	0.210↑	0.190	0.260↑	0.260↑	0.260↑	0.260↑	0.260↑	0.260↑	0.260↑
FACEFOUR	**0.080**	0.182↑	0.182↑	0.148	0.125	0.102	**0.080**	0.216↑	0.182↑	0.182↑	0.511↑	0.250↑	0.114	0.136	0.148	0.261↑	0.239↑
FISH	0.217↑	0.177↑	0.171	0.166	0.160	**0.103**	0.177↑	0.217↑	0.177↑	0.171	0.503↑	0.457↑	0.397↑	0.349↑	0.343↑	0.385↑	0.394↑
GUNPOINT	0.047	0.093↑	0.067↑	0.087↑	0.067↑	**0.020**	0.040	0.087↑	0.093↑	0.067↑	0.397↑	0.387↑	0.393↑	0.286↑	0.213↑	0.207↑	0.273↑
LIGHTING2	0.180	**0.115**	0.148	0.131	0.131	0.246	0.148	0.246	**0.115**	0.148	0.377↑	0.377↑	0.377↑	0.377↑	0.377↑	0.377↑	0.377↑
LIGHTING7	0.356	0.315	0.247	0.274	**0.233**	0.274	0.288	0.425↑	0.315	0.247	0.399↑	0.384↑	0.465↑	0.410↑	0.383↑	0.410↑	0.384↑
MEDICALIMAGES	0.316↑	0.275	0.261	0.275	0.266	**0.259**	0.284	0.316↑	0.275	0.261	0.611↑	0.533↑	0.575↑	0.564↑	0.536↑	0.563↑	0.486↑
OLIVEOIL	**0.133**	**0.133**	**0.133**	**0.133**	**0.133**	**0.133**	**0.133**	**0.133**	**0.133**	**0.133**	0.367↑	0.433↑	0.167	0.167	**0.133**	0.167	0.167
OSULEAF	0.467↑	0.430↑	0.421↑	0.368	0.343	**0.306**	0.463↑	0.483↑	0.430↑	0.421↑	0.794↑	0.842↑	0.636↑	0.521↑	0.498↑	0.603↑	0.507↑
PLANE	0.029	**0.000**	**0.000**	**0.000**	**0.000**	0.009	0.009	0.038↑	**0.000**	**0.000**	0.152↑	0.162↑	0.009	0.009	0.009	0.029	0.029
SWEDISHLEAF	0.206↑	0.216↑	0.203↑	0.222↑	0.211↑	0.266↑	**0.126**	0.211↑	0.216↑	0.203↑	0.550↑	0.511↑	0.317↑	0.326↑	0.296↑	0.309↑	0.311↑
SYMBOLES	0.099↑	0.055	0.061	0.050	0.059	0.051	0.099↑	0.100↑	0.055	0.061	0.109↑	0.084↑	0.063	0.054	**0.044**	0.067	0.072
TRACE	0.240↑	**0.000**	**0.000**	**0.000**	**0.000**	0.010	0.040↑	0.240↑	**0.000**	**0.000**	0.060↑	0.028	0.080↑	0.020	**0.000**	0.025	0.060↑
TWOPATTERNS	0.355↑	**0.000**	0.025↑	**0.000**	0.078↑	0.010	0.110↑	0.532↑	**0.000**	0.073↑	0.093↑	0.090↑	0.060↑	0.070↑	0.060↑	0.030↑	0.060↑
UMD	0.194↑	0.118↑	**0.014**	0.118↑	**0.014**	0.035	0.042	0.194↑	0.132↑	**0.014**	0.446↑	0.458↑	0.451↑	0.430↑	0.386↑	0.347↑	0.393↑
YOGA	0.317	0.367	0.337	0.317	0.343	0.323	**0.310**	0.317	0.367	0.337	0.430↑	0.530↑	0.483↑	0.467↑	0.333	0.400↑	0.433↑

Table 4.5: Classification error rate (k=1)

	k-nearest centroid				k-nearest weighted centroid		
	NLAAF	PSA	CWRT	DBA			
	DTW	DTW	DTW	DTW	WDTW	wk_{DTAK}	wk_{GDTW}
ADIAC	0.567↑	0.527	0.519	0.517	**0.465**	0.471	0.496
BEEF	0.600	**0.533**	0.600	0.567	**0.533**	0.600	0.600
BME	0.360	0.360	0.360	0.360	**0.333**	0.353	0.360
CAR	0.583↑	0.467	0.400	0.367	**0.337**	0.383	0.400
CBF	0.106↑	0.220↑	0.036	0.036	**0.031**	0.038	0.033
CC	0.113	0.377	0.037	**0.027**	**0.027**	0.087	0.110
COFFEE	**0.464**	**0.464**	**0.464**	**0.464**	**0.464**	**0.464**	**0.464**
CONSSEASON	0.350↑	0.389↑	0.172↑	0.133↑	**0.022**	0.244↑	0.311↑
ECG200	**0.260**	**0.260**	**0.260**	**0.260**	**0.260**	**0.260**	**0.260**
FACEFOUR	0.511↑	0.250↑	**0.114**	0.136	0.148	0.261↑	0.239↑
FISH	0.503↑	0.457↑	0.397	0.349	**0.343**	0.385	0.394
GUNPOINT	0.397↑	0.387↑	0.393↑	0.286	0.213	**0.207**	0.273
LIGHTING2	**0.377**	**0.377**	**0.377**	**0.377**	**0.377**	**0.377**	**0.377**
LIGHTING7	0.399	0.384	0.465	0.410	**0.383**	0.410	0.384
MEDICALIMAGES	0.611↑	0.533	0.575↑	0.564↑	0.536	0.563↑	**0.486**
OLIVEOIL	0.367↑	0.433↑	0.167	0.167	**0.133**	0.167	0.167
OSULEAF	0.794↑	0.842↑	0.636↑	0.521	**0.498**	0.603↑	0.507
PLANE	0.152↑	0.162↑	**0.009**	**0.009**	**0.009**	0.029	0.029
SWEDISHLEAF	0.550	0.511	0.317	0.326	**0.296**	0.309	0.311
SYMBOLES	0.109↑	0.084↑	0.063	0.054	**0.044**	0.067	0.072↑
TRACE	0.060	0.028	0.080↑	0.020	**0.000**	0.025	0.060↑
TWOPATTERNS	0.093↑	0.090↑	0.060↑	0.070↑	0.060↑	**0.030**	0.060↑
UMD	0.446	0.458	0.451	0.430	0.386	**0.347**	0.393
YOGA	0.430↑	0.530↑	0.483↑	0.467↑	**0.333**	0.400	0.433↑

Table 4.6: Nearest centroid classification error rate (k=1)

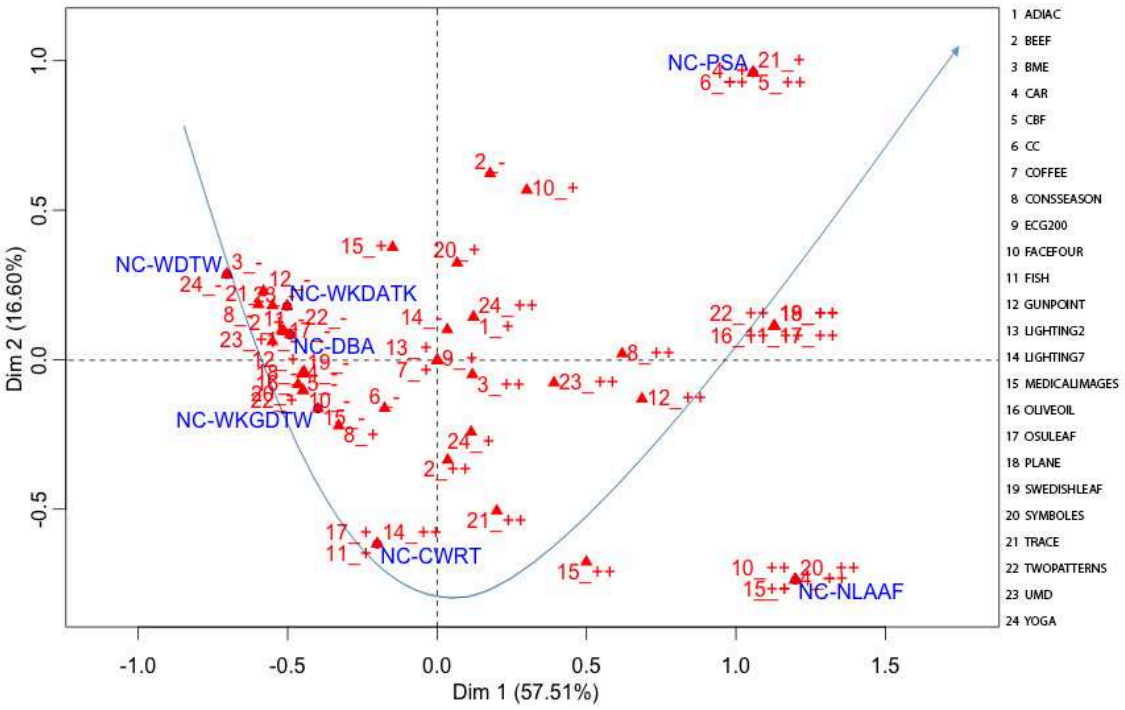


Figure 4.9: Global comparison of the nearest centroid classification error rate (k=1)

Complexity and time considerations

Table 4.7 shows the time consumption for different classification methods. In time comparison, we ignore the classification methods using k-NN with standard Gaussian kernel (RBF) and the k-NN (with Euclidean). While these methods use Euclidean, they are not comparable with the other methods under time warp. As one can note, the nearest centroid classifier using CWRT and DBA are the fastes classification methods, similar to the proposed nearest weighted centroid classification methods. The slight difference is related to the weight computation for each centroid. Whereas we significantly improve the accuracy in the proposed nearest weighted centroid classifiers, one can ignore this non-significant difference in run time. As we mentioned before, the main drawback of NLAAF and PSA averaging methods lie in the growth of their resulting lengths. While nearest centroid classification methods normally should be faster than the nearest neighbor ones, the nearest centroid classifiers using NLAAF or PSA are highly time-consuming (particulary for large datasets), because of the progressive increase of the centroid length during the pairwise combination process.

Figure 4.10 shows the multiple correspondence analysis of classification time consumption to compare globally the classification time consumption of different methods. The first group is defined by the k-NN standard Gaussian kernel (RBF) and k-NN (with Euclidean), followed by the nearest neighbor classifiers using CWRT, DBA, WDTW, $\mathrm{WK}_{\mathrm{DTAK}}$ and $\mathrm{WK}_{\mathrm{GDTW}}$. As you can see in the figure 4.10, k-NN using kernel $\kappa_{\mathrm{DTAK}_{sc}}$ is next in ranking based on the time consumption, followed by κ_{TGA}, k-NN using DTW_{sc} and kernel $\kappa_{\mathrm{GDTW}_{sc}}$. The third group is contains the k-nearest neighbor classifiers using DTW and kernels. The last group, the classification approaches which use centroid estimation with NLAAF and PSA yield the lowest performances.

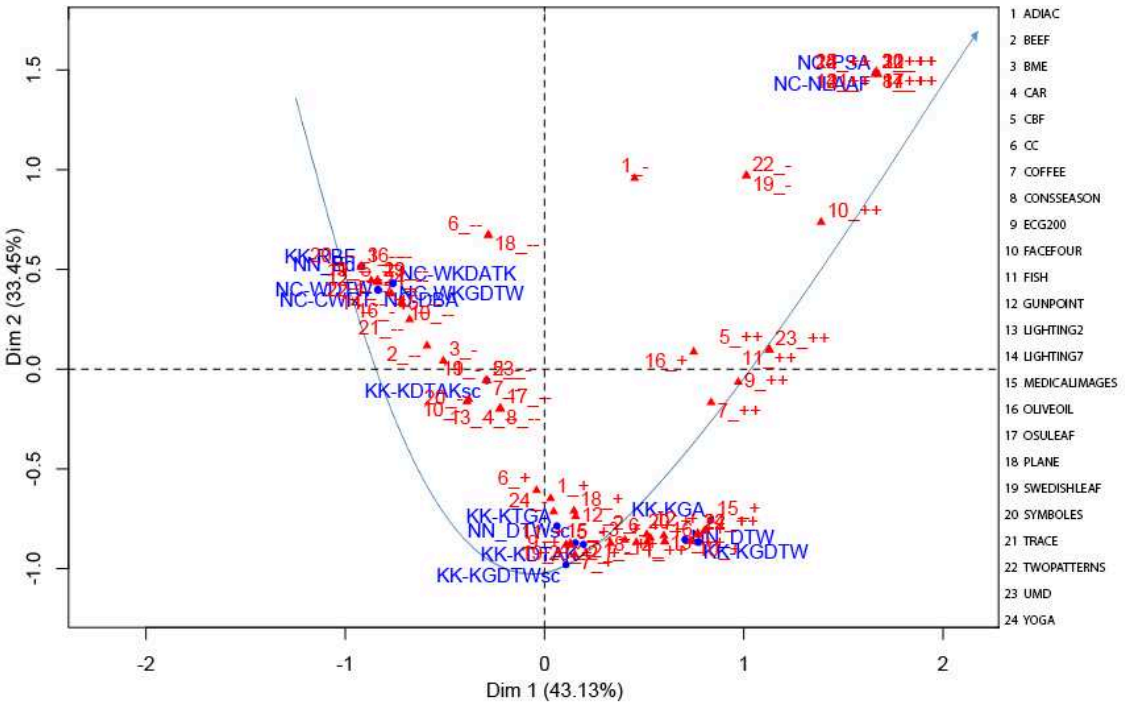


Figure 4.10: Global comparison of the classification Time consumption (k=1)

	k-NN kernel							k-NN (standard)			k-nearest centroid NLAAF	k-nearest centroid PSA	k-nearest centroid CWRT	k-nearest centroid DBA	k-nearest weighted centroid		
	RBF	k_{GDTW}	$k_{GDTW_{sc}}$	k_{DTAK}	$k_{DTAK_{sc}}$	k_{GA}	k_{TGA}	e_d	DTW	DTW_{sc}	DTW	DTW	DTW	DTW	WDTW	wk_{DTAK}	wk_{GDTW}
ADIAC	1.43	554.8↑	449.8↑	358.6↑	303.3↑	547.6↑	354.8↑	0.23	581.4↑	425.1↑	231.2↑	246.5↑	133.1	**131.2**	142.8	144.7	147.9
BEEF	0.05	28.3↑	21.9	23.7↑	**13.4**	29.5↑	22.9	0.09	26.8↑	21.4	79.2↑	88.2↑	13.8	13.9	15.0	15.4	16.1
BME	0.09	9.2↑	7.0↑	6.2↑	4.5↑	8.1↑	6.0↑	0.09	8.9↑	6.9↑	10.8↑	12.7↑	**3.3**	**3.3**	3.4	3.5	3.5
CAR	0.12	156.6↑	136.4↑	134.9↑	77.2↑	188.2↑	135.7↑	0.03	161.1↑	129.4↑	912.3↑	1049.8↑	44.9	**44.7**	45.9	46.0	47.2
CBF	0.54	64.1↑	46.7↑	43.6↑	31.8↑	49.8↑	39.7↑	0.31	55.7↑	45.5↑	50.4↑	55.3↑	**14.8**	14.9	15.0	15.2	15.3
CC	0.24	51.1↑	37.5↑	41.1↑	29.2↑	46.8↑	33.2↑	0.11	45.8↑	38.8↑	9.7↑	11.6↑	**2.4**	**2.4**	2.6	2.7	2.7
COFFEE	0.02	8.4↑	6.0↑	5.2↑	3.1↑	7.2↑	6.9↑	0.01	8.1↑	6.9↑	8.2↑	7.9↑	**0.7**	**0.7**	**0.7**	0.8	0.9
CONSSEASON	0.17	82.3↑	78.9↑	58.6↑	36.5↑	79.9↑	54.3↑	0.06	81.4↑	61.5↑	358.7↑	378.9↑	**2.2**	**2.2**	2.4	2.5	2.5
ECG200	0.07	12.5↑	8.9↑	9.7↑	5.9↑	11.8↑	7.9↑	0.03	11.5↑	8.8↑	9.4↑	10.4↑	**0.2**	**0.2**	**0.2**	**0.2**	0.2
FACEFOUR	0.06	35.8↑	25.3↑	22.4↑	**13.2**	40.9↑	23.9↑	0.02	33.0↑	24.7↑	49.8↑	54.0↑	15.6	15.7	16.9	17.1	17.2
FISH	0.72	898.4↑	786.2↑	584.9↑	362.7↑	955.1↑	710.7↑	0.09	865.1↑	664.1↑	1022.2↑	1107.8↑	**106.7**	106.8	107.4	107.9	108.0
GUNPOINT	0.10	20.9↑	16.3↑	15.3↑	11.6↑	27.3↑	16.7↑	0.04	20.4↑	15.8↑	39.3↑	41.2↑	2.2	**2.1**	2.2	2.3	2.3
LIGHTING2	0.14	228.4↑	187.6↑	191.8↑	108.8↑	297.2↑	161.6↑	0.03	225.1↑	174.2↑	7682.4↑	9666.5↑	**11.2**	11.3	12.6	12.7	12.8
LIGHTING7	0.11	68.1↑	50.2↑	50.9↑	38.8↑	76.8↑	42.7↑	0.03	70.4↑	49.8↑	132.2↑	155.6↑	**16.9**	**16.9**	17.2	17.8	17.7
MEDICALIMAGES	1.49	380.8↑	277.6↑	282.5↑	197.9↑	448.9↑	250.2↑	0.31	375.1↑	271.4↑	853.2↑	959.8↑	20.8	**20.7**	21.6	22.0	22.2
OLIVEOIL	0.04	47.3↑	37.9↑	32.2↑	20.2	45.8↑	32.7↑	0.01	45.2↑	32.0↑	26.2	29.3↑	**16.2**	**16.2**	16.4	16.5	16.5
OSULEAF	1.03	1318.3↑	894.7↑	775.8↑	563.3↑	1244.9↑	845.2↑	0.14	1238.8↑	877.6↑	10877.3↑	20858.4↑	**106.2**	107.7	110.1	111.5	111.8
PLANE	0.12	37.5↑	21.9↑	29.2↑	22.1↑	37.0↑	27.4↑	0.03	27.6↑	19.2↑	8.6↑	8.5↑	**4.7**	**4.7**	4.8	4.9	5.0
SWEDISHLEAF	1.93	687.9↑	496.7↑	470.7↑	293.2↑	737.6↑	487.4↑	0.28	640.2↑	503.4↑	255.7↑	272.2↑	40.9	**40.8**	41.6	41.7	41.8
SYMBOLES	0.76	581.4↑	411.9↑	328.8↑	292.3↑	590.6↑	420.3↑	0.25	555.0↑	410.2↑	714.0↑	880.6↑	**256.1**	256.3	258.4	262.1	264.7
TRACE	0.13	94.4↑	72.2↑	74.4↑	50.2↑	95.4↑	61.8↑	0.04	101.2↑	77.8↑	202.3↑	283.9↑	**10.3**	**10.3**	10.4	10.5	10.5
TWOPATTERNS	0.38	89.2↑	63.4↑	67.8↑	43.6↑	106.1↑	57.9↑	0.13	84.2↑	61.4↑	48.1↑	41.2↑	**7.6**	7.7	7.9	8.0	8.1
UMD	0.12	15.1↑	10.8↑	9.8↑	6.2↑	12.6↑	9.2↑	0.04	14.3↑	10.8↑	12.3↑	13.9↑	**2.8**	**2.8**	3.0	3.0	3.0
YOGA	0.27	196.6↑	161.5↑	137.4↑	95.9↑	224.6↑	158.2↑	0.08	230.9↑	168.2↑	285.9↑	377.7↑	33.6	**33.4**	35.9	36.9	36.5

Table 4.7: Classification time consumption (k=1)

Classification goodness criteria

Lastly, for a global comparison considering both classification error rate and run time index, we define here a goodness criteria as a ratio of "*fastness*" to "*accuracy*". The "*fastness*" (normalized time consumption), should be lower and the "*accuracy*" (1 - error rate), should be higher.

$$\text{CLASSIFICATION GOODNESS} = \frac{\text{fastness}}{\text{accuracy}}$$

$$= \frac{\text{normalized run time}}{1 - \text{error rate}}$$

Table 4.8 presents the classification average ranking on different methods, according to the defined goodness criteria. Note that, we ignore the k-NN classification based on Euclidean and kernel RBF, whereas they are not comparable with the classification methods which use time warp measures. The best performance belongs to the proposed nearest weighted centroid classifier with WDTW with the average rank of 1.81, followed by the nearest centroid classifier using using DBA, with the average rank of 2.48, followed by the nearest weighted centroid classifier using WK_{DTAK}. The classifiers which use nearest centroid with NLAAF or PSA, and nearest neighbor using kernels κ_{GDTW} or κ_{GA} yield the lowest performances according to the global average ranking.

k-nearest neighbor kernel						k-NN standard		k-nearest centroid				k-nearest weighted centroid		
k_{GDTW}	$k_{\text{GDTW}_{sc}}$	k_{DTAK}	$k_{\text{DTAK}_{sc}}$	k_{GA}	k_{TGA}	DTW	DTW_{sc}	NLAAF	PSA	CWRT	DBA	WDTW	wk_{DTAK}	wk_{GDTW}
12.83	9.48	8.58	5.70	12.70	8.67	11.99	8.69	12.46	12.92	3.43	2.48	**1.81**	3.35	4.18

Table 4.8: Classification average ranking (k=1)

4.3.3 A closer look at the centroids

We finally visualize here some of the centroids obtained by the different methods and compare their shape to the one of the time series they represent. Figures 4.14 to 4.19 display the centroids obtained by the different methods. The first line shows some randomly selected time series of the averaged sets, the second line the centroids provided by the alternative methods and the last line the centroids obtained by WDTW, WK_{DTAK} and WK_{GDTW}. Finally, the estimated weights are shown below the corresponding centroids. To assess the weights obtained, we rely on the following observation: a uniform weight distribution is expected for time series that behave similarly within classes, as the centroid's elements are equivalently shared within the class, whereas a non-uniform distribution is awaited for time series that behave differently within classes, as higher weights are assigned to shared centroid's elements. Thus, it is possible to assess the relevance of the estimated weights by examining in which cases their distribution is uniform, and in which cases it is not. To do so, we perform for each

times series collection a $\mathcal{X}^2$-test (at 5% risk) measuring the distance between the estimated weights and a uniform model. A uniform weights distribution is accepted for a p-value greater than 0.05 (*e.g.* CC-"Cyclic"), otherwise rejected (*e.g.* UMD-"Down").

As one can note, for standard datasets, almost all the approaches succeed in obtaining centroids more or less similar to the initial time series (given at the first line). For instance, in Figures 4.15 for the classes "Cyclic" of the CC dataset. However, we observe generally less representative centroids for NLAAF and PSA with a drastically large centroid's length of about 10^3 elements vs. 10^2 for the other methods. For UMD data, Figure 4.16 shows that the WDTW, WK_{DTAK} and WK_{GDTW} provide the most representative centroid with the highest weights successfully assigned to the begin and end regions that characterize the UMD dataset.

To have a closer look at the ability of the proposed method in centroid estimation, here we justify our claim on two complex and completely noisy datasets. The SPIRAL data, proposed in [ZT09], consists of 3-D spatio-temporal time series (2-D spatial and 1-D temporal) generated from latent time series:

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{U}_i^T(\mathbf{z} + \mathbf{b}_i \mathbf{1}_l^T)\mathbf{M}_i \\ \mathbf{e}_i^T \end{bmatrix} \in \mathbb{R}^{3*n_i}$$

where the canonical time series $\mathbf{z} \in \mathbb{R}^{2*l}$ is a curve in two dimensions (x, y). $\mathbf{U}_i \in \mathbb{R}^{2*2}$ and $\mathbf{b}_i \in \mathbb{R}^2$ are randomly generated projection matrix and translation vector respectively. The binary matrix $\mathbf{M}_i \in \{0,1\}^{l*n_i}$ is generated by randomly choosing $n_i \leq l$ columns from $\mathbf{I}_l$ for temporal distortion. The spatial dimension $\mathbf{e}_i \in \mathbb{R}^{n_i}$ is generated with zero-mean Gaussian noise. The latent time series $\mathbf{z}$ and three sample generated time series are visualized in Figure 4.11.

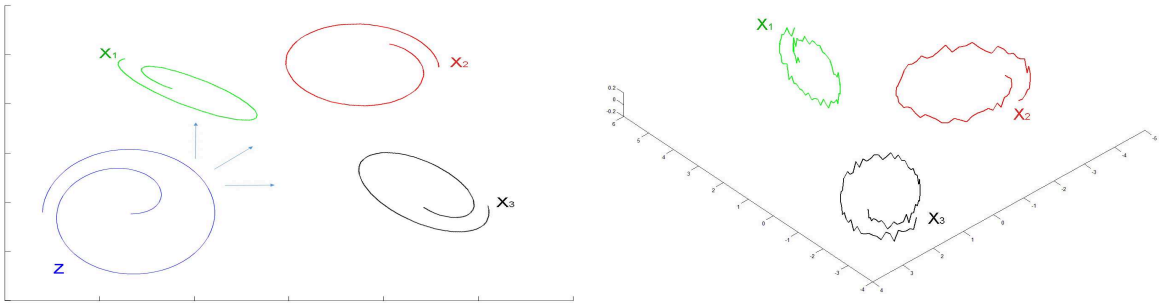


Figure 4.11: Latent curve $\mathbf{z}$ and three induced instances $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ without noise (left), and with noise e_i (right) - SPIRAL dataset

SPIRAL2 extends SPIRAL data to more challenging time series that are highly noisy and globally behave differently while sharing a 3 dimensional latent time series that may appear randomly at different time stamps. The latent time series $\mathbf{z}$ and three generated time series are visualized in Figure 4.12. Figure 4.13 shows the progression of their x, y and e dimensions over time.

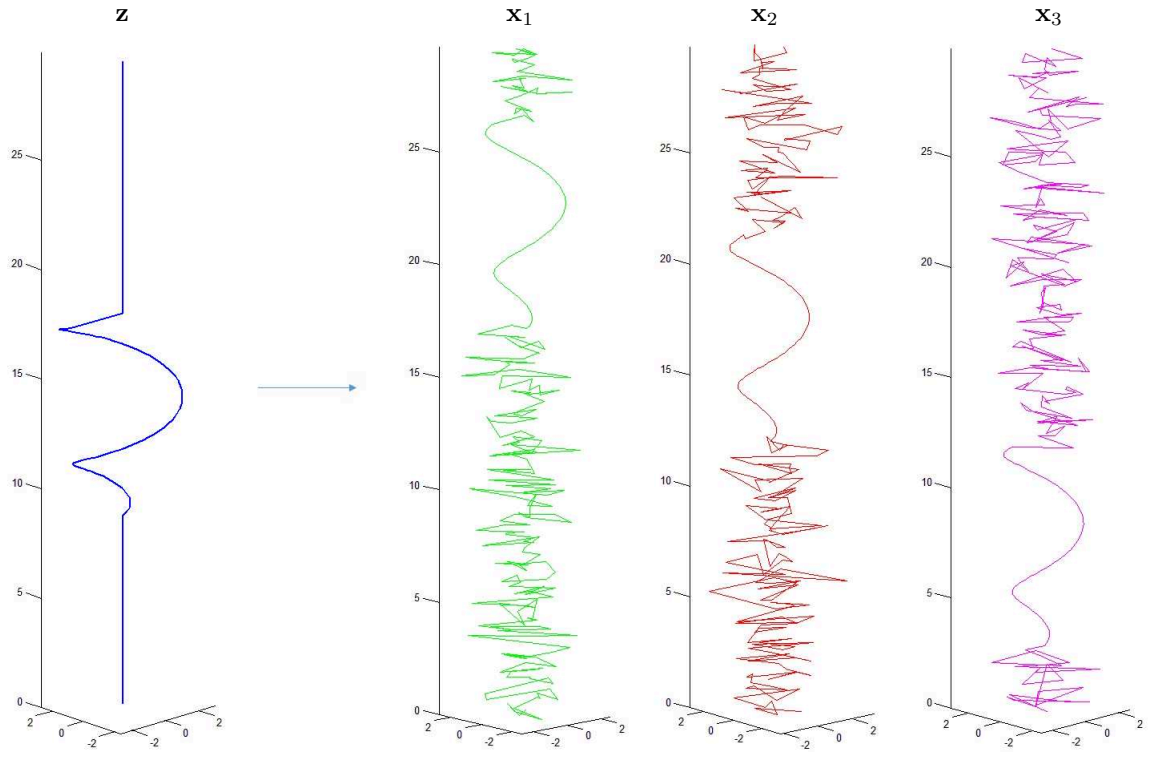


Figure 4.12: Latent curve $\mathbf{z}$ and three induced instances $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ sharing local characteristics for the SPIRAL2 dataset

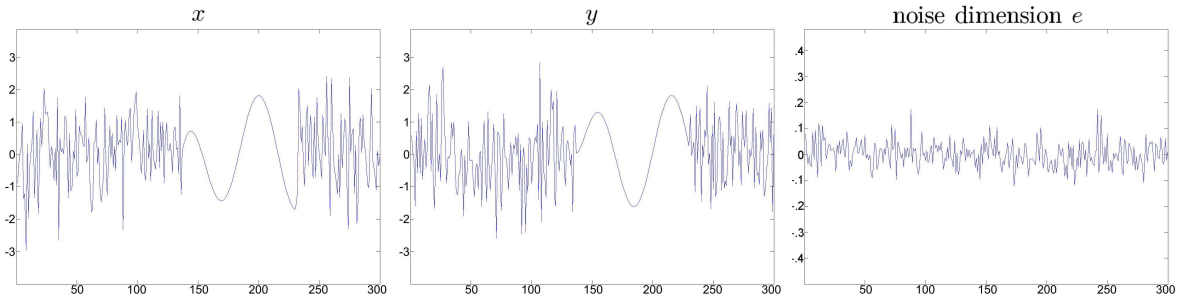


Figure 4.13: SPIRAL2: Progression of the 2-D spatial coordinates (x, y) and the noise dimension e over time for the SPIRAL2 dataset

According to Figure 4.17, on SPIRAL data, only WDTW, WK_{DTAK} and WK_{GDTW} succeed in estimating centroids with an exact spiral shape, as well as providing an effective uniform weighting, as SPIRAL relies on time series of a similar global behavior. For the more complex SPIRAL2 collection, Figure 4.18 shows the ability of the proposed methods (WDTW, WK_{DTAK} and WK_{GDTW}), to circumvent the noise problem and to reveal the locally shared signature. For example, in Figure 4.19, we can see clearly for the dimension x of SPIRAL2 that (a) the initial hidden signature is well revealed by the estimated centroids, (b) the noise is removed, and (c) the corresponding region is highly weighted as expected.

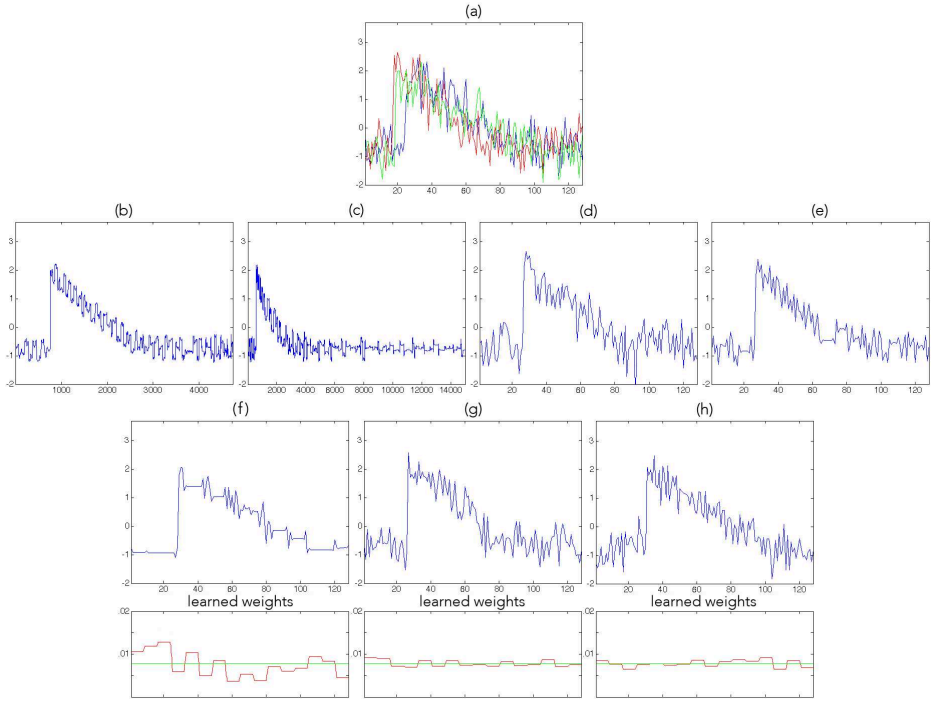


Figure 4.14: CBF-"Funnel" centroids: (a) Initial time series, (b) NLAAF, (c) PSA, (d) CWRT, (e) DBA, (f) WDTW, (g) WK_{DTAK}, (h) WK_{GDTW}

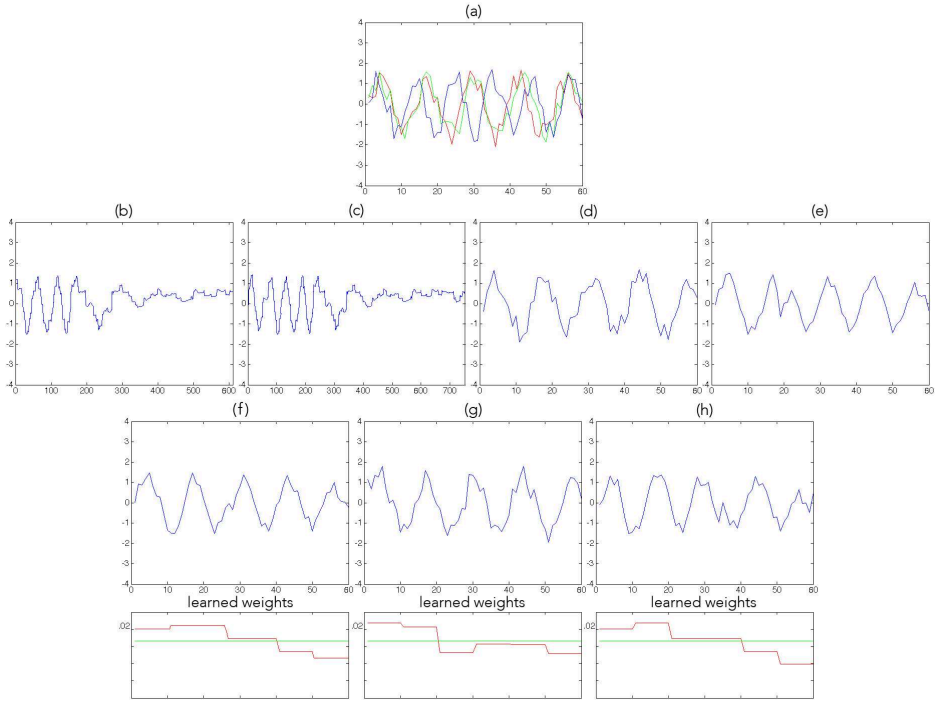


Figure 4.15: CC-"Cyclic" centroids: (a) Initial time series, (b) NLAAF, (c) PSA, (d) CWRT, (e) DBA, (f) WDTW, (g) WK_{DTAK}, (h) WK_{GDTW}

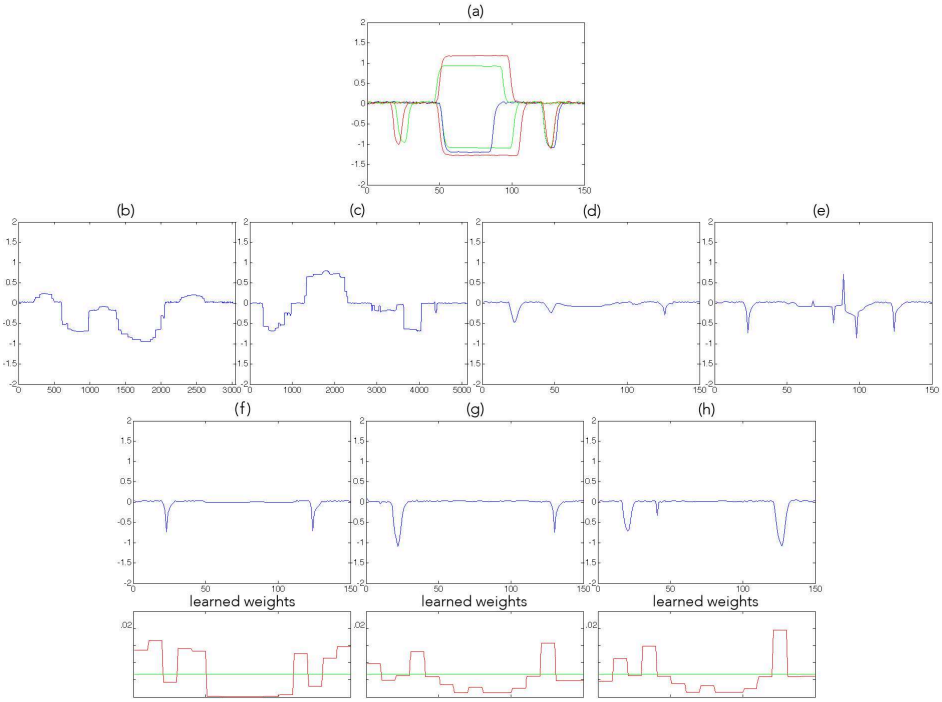


Figure 4.16: UMD-"Down" centroids: (a) Initial time series, (b) NLAAF, (c) PSA, (d) CWRT, (e) DBA, (f) WDTW, (g) WK_{DTAK}, (h) WK_{GDTW}

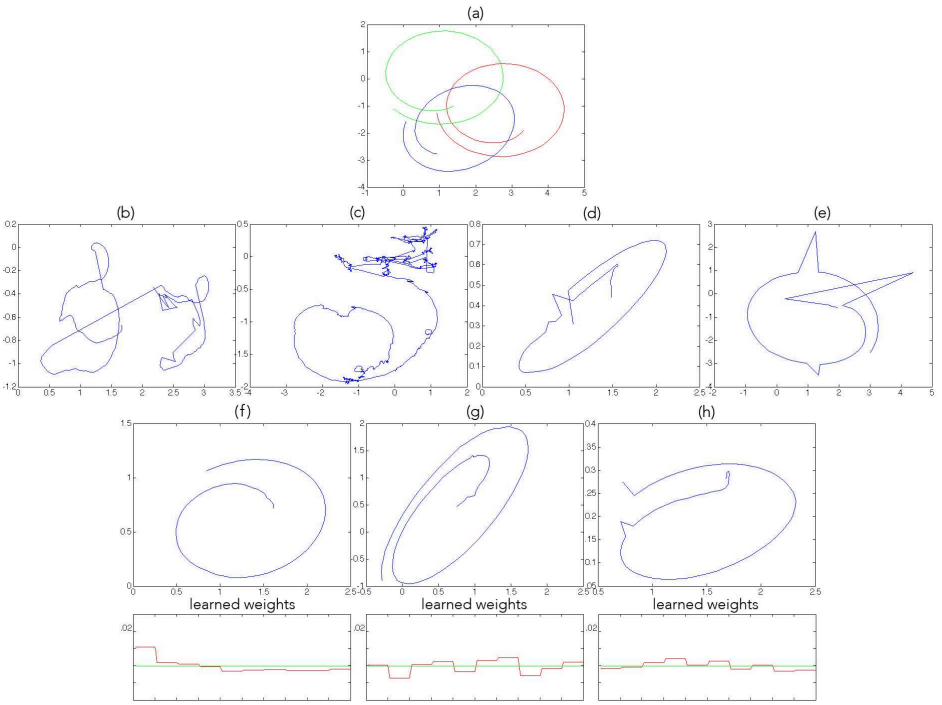


Figure 4.17: SPIRAL centroids: (a) Initial time series, (b) NLAAF, (c) PSA, (d) CWRT, (e) DBA, (f) WDTW, (g) WK_{DTAK}, (h) WK_{GDTW}

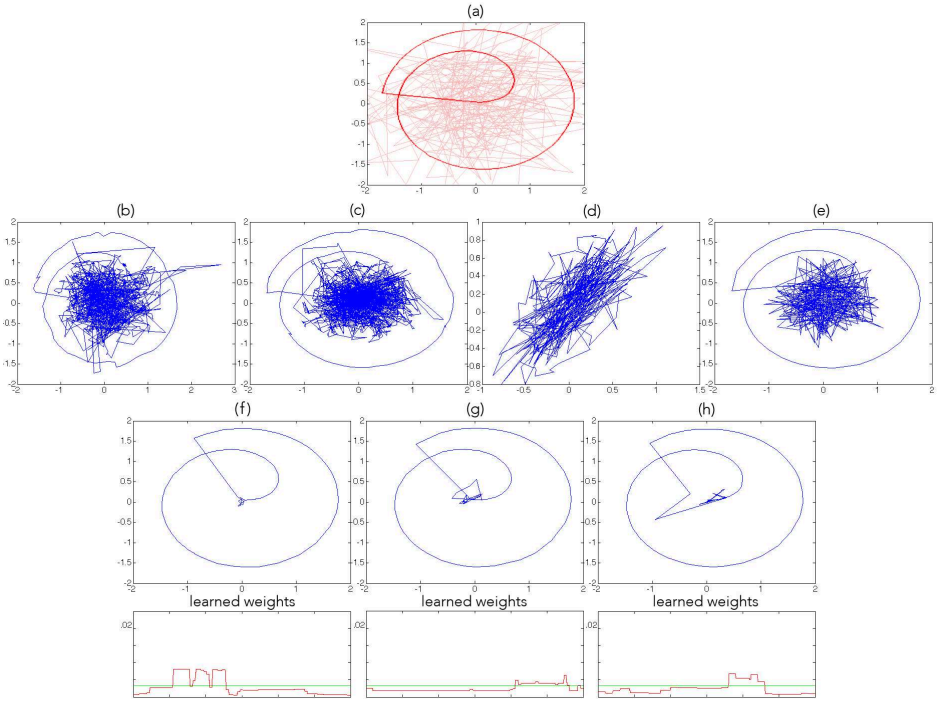


Figure 4.18: SPIRAL2 centroids: (a) Initial time series, (b) NLAAF, (c) PSA, (d) CWRT, (e) DBA, (f) WDTW, (g) WK_{DTAK}, (h) WK_{GDTW}

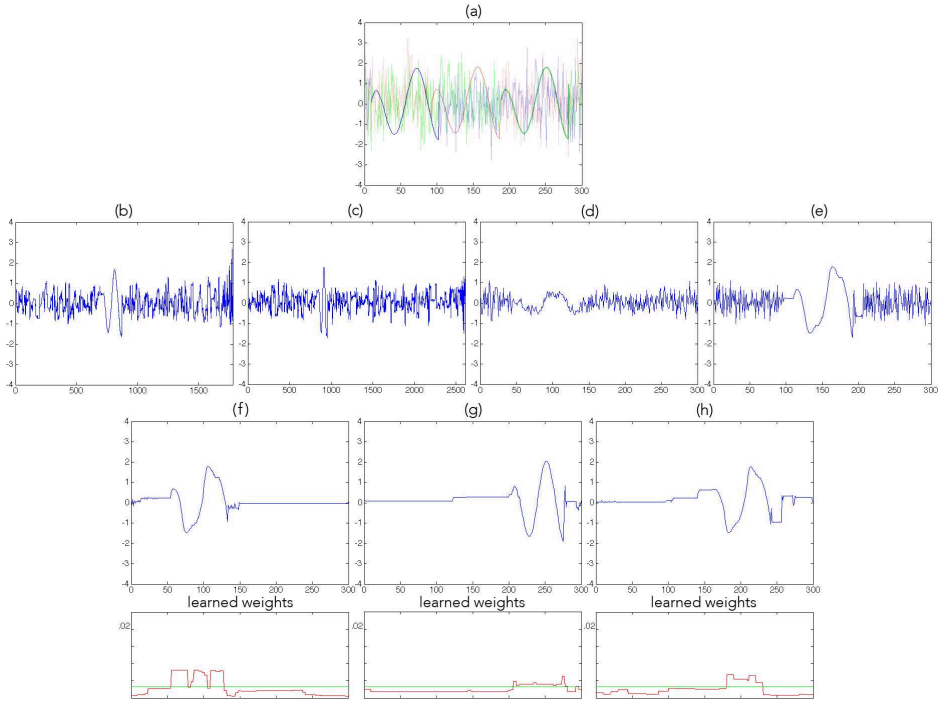


Figure 4.19: SPIRAL2-X centroids: (a) Initial time series, (b) NLAAF, (c) PSA, (d) CWRT, (e) DBA, (f) WDTW, (g) WK_{DTAK}, (h) WK_{GDTW}

Figures 4.20 and 4.21 complete these results and show the progressive estimation of the centroid and weights through the first iterations of WDTW and WK_{DTAK}, respectively. These figures show the ability of the proposed methods to remove the noise and to capture the shared signature fastly during the first few iterations of the estimation process. The first row shows the initial centroid (left) and the initial weights (right), which are from uniform distribution. Each row shows one iteration, and finally, the last row, represents the estimated centroid and its weight vector.

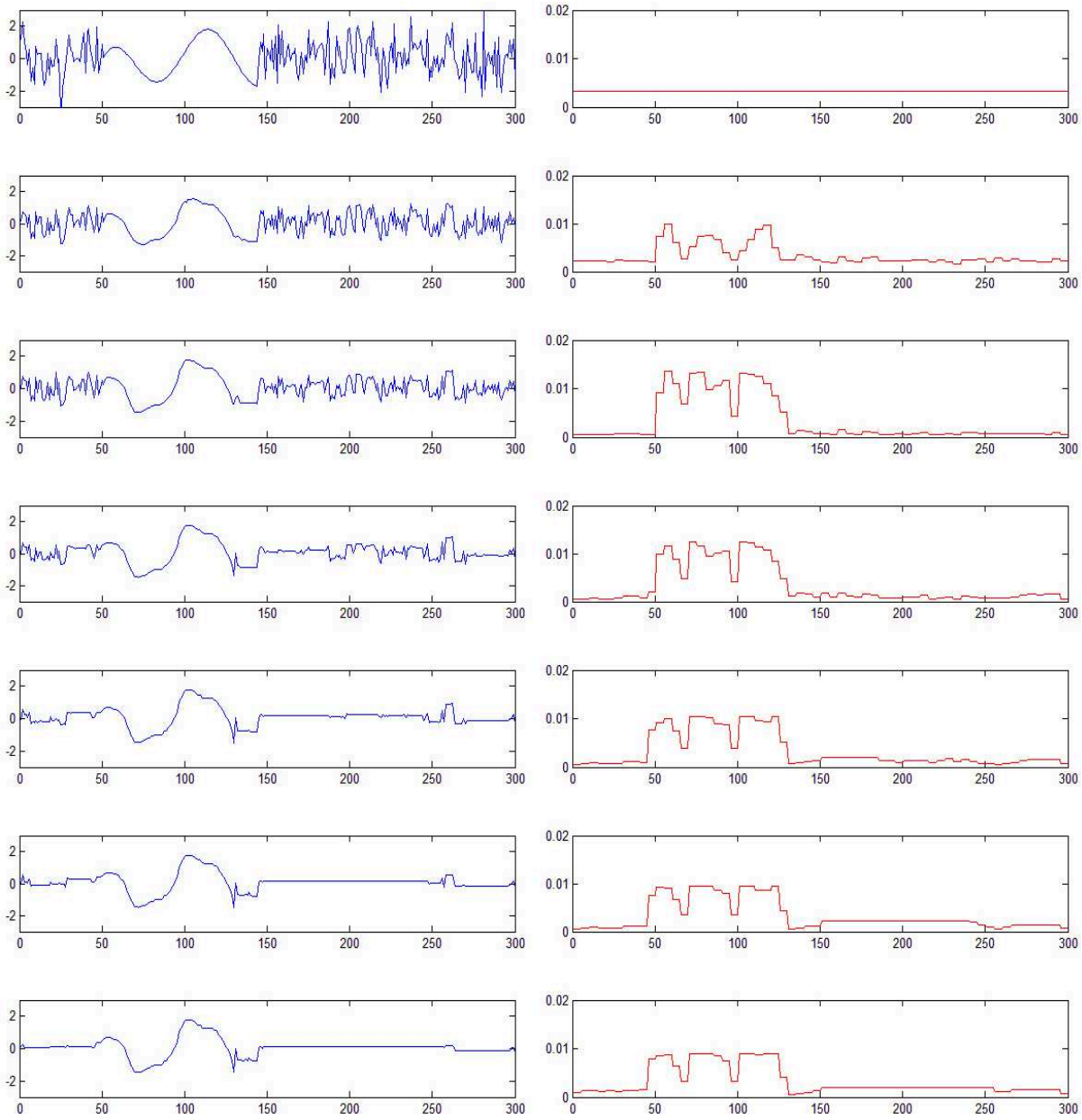


Figure 4.20: SPIRAL2-X centroids: centroid (left) and weight (right) estimation through the first iterations of WDTW

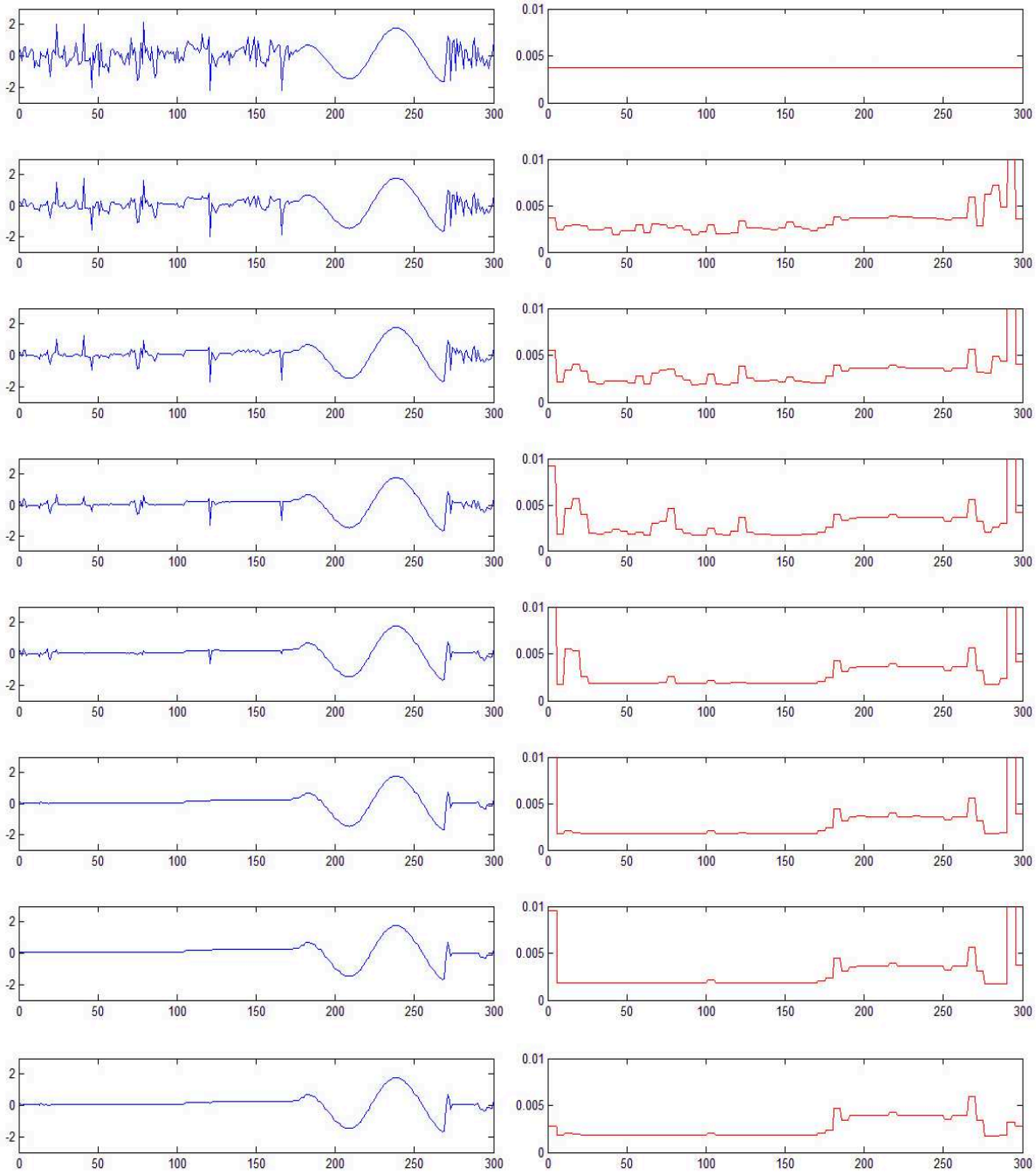


Figure 4.21: SPIRAL2-X centroids: centroid (left) and weight (right) estimation through the first iterations of $\mathrm{WK}_{\mathrm{DTAK}}$

4.4 Discussion

From Table 4.3, we can see that the generalized k-means clustering leads to the highest Rand indexes, where the best scores (indicated in bold) are reached by WDTW for almost all

datasets. To have a global view of clustering results, Figure 4.22 shows the average Rand index values of all datasets, for each clustering method. The black color illustrates the best and white color presents the worst one. From Table 4.4, the results reveal WDTW the fastest methods under time warp and k-means using PSA the slowest one. The Generalized k-means using WK_{DTAK} and WK_{GDTW} appears slightly slower that WDTW; this is due to the gradient ascent research part used in the former.

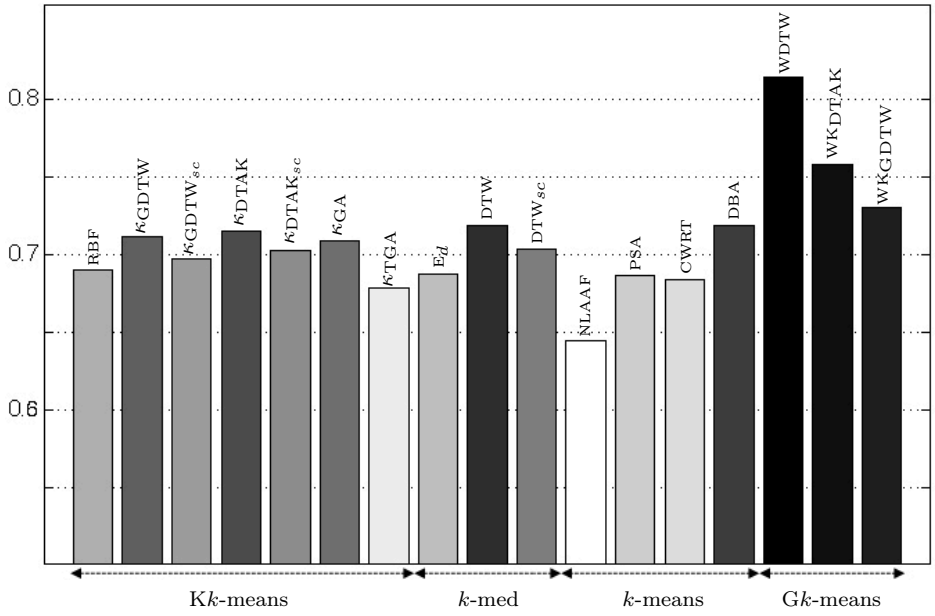


Figure 4.22: Comparison of average Rand index values

To evaluate the relevance of the estimated weights, we performed a $\mathcal{X}^2$-test[10] (at 5% risk, the significance level of the hypothesis test) measuring the distance between the estimated weights and a uniform model. The p-value is the probability of observing a test statistic more extreme than the observed value under the null hypothesis. Table 4.9 allows one to corroborate the characteristics of some sample datasets for WDTW, WK_{DTAK} and WK_{GDTW}: the uniform hypothesis is accepted for all standard datasets as well as for SPIRAL dataset (*i.e.* on all the datasets in which time series behave similarly within the classes) as p-values are greater than 0.05, and strongly rejected for the complex datasets like BME, UMD and SPIRAL2, in which time series behave differently with in the classes, as p-values are lower than 0.05.

Regarding the classification results (Table 4.8), for all datasets, the highest classification goodness scores are obtained by a nearest centroid classifier using WDTW, followed by DBA and WK_{DTAK}.

[10] chi2gof function in Matlab 2015b, which returns the p-value of the hypothesis test

	p-values		
	WDTW	WKDTAK	WKGDTW
BME	$\simeq 0.00$	$\simeq 0.00$	$\simeq 0.00$
CBF	1	1	1
CC	1	1	1
UMD	$\simeq 0.00$	$\simeq 0.00$	$\simeq 0.00$
SPIRAL	1	1	1
SPIRAL2	$\simeq 0.00$	$\simeq 0.00$	$\simeq 0.00$

Table 4.9: p-value ($\mathcal{X}^2$-test: uniformity distribution test)

Lastly, the qualitative evaluations (Figures 4.14 to 4.19, consisting in the visualization and comparison of the centroids obtained by different methods) show the effectiveness of the proposed approaches to provide accurate time series averaging for standard and complex datasets.

Conclusion and perspectives

The DTW and kernel-DTW are among the most frequently used metrics for time series in several domains as signal processing, data mining or machine learning. However, for clustering time series, approaches are generally limited to the k-medoids or kernel k-means to circumvent centroid estimation under DTW and the tricky multiple temporal alignment problem. This work introduces a generalized k-means-based clustering algorithm for temporal data under extended time warp measures. For this, we propose i) an extension of the common time warp measures and ii) an accurate, fast and efficient solution to the problem of time series averaging, under the extended time warp measures, that captures local temporal features. Additionally, we introduce an extension of time series averaging to kernel-DTW metric as it constitutes a crucial metric for many kernel approaches; Yields a fast method to compute the centroid of time series. Accordingly, we generalize the averaging problem to time series of distinct behaviors that share local characteristics by estimating a weight vector, through several warping function. The integration of a weight vector permits us to weigh time stamps differently and indicate the importance of each time stamp. Lastly, we use the proposed centroid estimation and generalized k-means clustering approach in the classification context. In this way, to speed up the nearest neighbor classification, we suggest a fast accurate nearest weighted centroid classifier.

The efficiency of the proposed centroid estimation solution for classification and clustering algorithms is analyzed on a wide range of public standard and challenging datasets, which are non-spherical, non-well-isolated and (/or) linearly non-separable. The experimental validation is based on standard datasets in which time series share similar behaviors within classes, as well as on more complex datasets exhibiting time series that share only local characteristics, that are multidimensional and noisy.

The approaches are compared through several criteria. For quantitative evaluation, we consider a Rand index, a goodness clustering criterion, and the space and time requirements, as well as the classification error rate as a classification goodness. On the other hand, the qualitative one consists in the visualization and comparison of the centroids obtained by different methods, and the relevance of the weights estimated. In addition, we propose here to investigate the performance of the different methods on the different datasets through a multiple correspondence analysis that highlights the relationships between the methods and the datasets, indicates which method performs well or badly on which datasets, and suggests a global ranking of the different methods. Both the quantitative evaluation and the qualitative one demonstrate the effectiveness of the proposed approaches to being faster and more accurate, which outperform the other methods, in the context of averaging, classification and clustering.

In particular, the approaches introduced in this study have the following characteristics:

- The centroids obtained are more representative of the set, than the centroids obtained by other methods;
- The weights associated to the centroids reflect the elements that are shared locally in a set of time series;
- The estimated centroids yield better clusters in the k-means-based clustering than the centroids obtained by other methods;
- The time and space requirements are lower in comparison with the time and space requirements of other methods.

Perspectives

Explore new temporal metrics and warping functions In this thesis, we focus on four temporal metrics: the dynamic time warping, the dynamic temporal alignment kernel, the Gaussian dynamic time warping and the global alignment kernel. It could be interesting to integrate other temporal metrics as well as different warping functions in our proposed framework to improve the results, specially for very challenging datasets. In this work, we consider only two different warping functions for each temporal metric. Even we are globally best method than the rest, one can explore more warping function, specially for challenging and noisy datasets with low values of Rand index.

Furthermore, in the context of time complexity, although we obtain the best run time in comparison with the alternatives, but still one can investigate for new warping functions, which leads to a suitable and closed-form solution, to circumvent the gradient-based optimization process in weight estimation steps and in result decrease the time consumption.

Lastly, the generalized k-means-based clustering on the $\mathrm{WK}_{\mathrm{DTAK}}$ and $\mathrm{WK}_{\mathrm{GDTW}}$ suggests a new promising alternative approach to kernel k-means, that encourages to deepen this issue for further temporal kernels and pre-image problem in future works.

Faster clustering/classification process by temporal segmentation For several large and challenging datasets as CHARACTER TRAJECTORIES[11], using the proposed weight and centroid estimation approach, one can simply distinguish the patterns of interest. Figure 4.23 shows the obtained segments of interest for some sample classes of CHARACTER TRAJECTORIES data. Pink color shows the segment of interest (instances with higher weights) for each sample character.

In this case, a second development is to decrease time and space complexities using a temporal segmentation of time series. The idea is to localize segments of interest, according

[11] UCI Machine Learning Repository

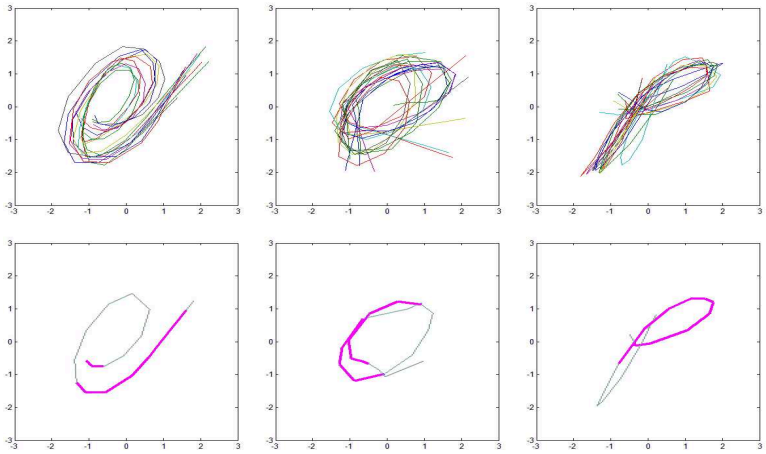


Figure 4.23: Three samples classes of CHARACTER TRAJECTORIES-"e", "o" and "p": the ground truth (top), the segments of interest (down)

to the estimated weight vector. By considering these segments, instead of whole time series, one can improve the time and space complexity in the context of comparison.

Study the potential of this thesis in other machine learning algorithms The next advancement will be evaluating the extended temporal measures in other algorithms. In this work, we studied time series averaging under time warp for clustering and classification. The next step is to evaluate them in an other classifier such as a decision tree or a support vector machine or extension of the proposed weighted temporal measures in the fuzzy c-means clustering (*i.e.* soft clustering) context, while in some application (*e.g.* bio-informatics, image analysis or marketing) each data point can belong to more than one cluster.

Other propositions to define the combined metric In this thesis, for each proposition, we considered only one single metric (*e.g.* WDTW) to compare time series. The idea of combined metrics can also be the other proposition to gain better results for some multidimensional challenging data sets. Investigate the combination of some behavior-based with value-based metrics, to deal with different dimension of data can be an interesting suggestion for future works.

Appendix A

Proof of Theorems 3.1 and 3.2

Let us first recall Theorem 3.1.

Theorem 3.1 *Let $f : [0,1] \to \mathbb{R}^+$ be the non-increasing function used in g, and let $\varphi : \mathbb{R}\times\mathbb{R} \to \mathbb{R}^+$ be the positive, real-valued function used in g. Let us furthermore define:*

$$\textbf{Condition 1:} \quad \forall a \in [-1,1],\ \forall x \in [0,1],\ \sum_{k=2}^{\infty} \frac{a^k}{k!} f^{(k)}(x) \geq 0$$
$$\textbf{Condition 2:} \quad \forall (c,c') \in \mathbb{R}^2, \forall x \in \mathbb{R}, \sum_{k=2}^{\infty} \frac{(c'-c)^k}{k!} \frac{\partial^k \varphi(x,c)}{\partial c^k} \geq 0$$

Then:

If Condition 1 holds, then $g(\boldsymbol{w}/\boldsymbol{c}, \boldsymbol{\Pi}^*)$ is pseudo-convex;
If Condition 2 holds, then $g(\boldsymbol{c}/\boldsymbol{w}, \boldsymbol{\Pi}^*)$ is pseudo-convex.

Proof: $g(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*)$ is pseudo-convex *iff*:

$$\forall\, \mathbf{w}, \mathbf{w}' \ \ \nabla g(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*)\,(\mathbf{w}' - \mathbf{w}) \geq 0 \implies g(\mathbf{w}'/\mathbf{c}, \mathbf{\Pi}^*) \geq g(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*)$$

Let

$$\mathcal{A}(\mathbf{X}, \mathbf{c}, \mathbf{\Pi}^*, t) = \sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} \varphi(x_{t'}, c_t)$$

where $\mathbf{\Pi}^*$ is the set of all known (and fixed) alignments for the time series $\mathbf{x}$ in $\mathbf{X}$. As only positive quantities are involved, $\mathcal{A}(\mathbf{X}, \mathbf{c}, \mathbf{\Pi}^*, t) \geq 0$. We have:

$$\text{(i)} \qquad g(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*) = \sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\varphi(x_{t'}, c_t),$$

$$\text{(ii)} \qquad \frac{\partial g(w_t/\mathbf{c}, \mathbf{\Pi}^*)}{\partial_{w_t}} = f'(w_t)\,\mathcal{A}(\mathbf{X}, \mathbf{c}, \mathbf{\Pi}^*, t),$$

$$\text{(iii)} \qquad \nabla g(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*)(\mathbf{w}' - \mathbf{w}) = \sum_{t=1}^{T} f'(w_t)(w'_t - w_t)\,\mathcal{A}(\mathbf{X}, \mathbf{c}, \mathbf{\Pi}^*, t),$$

$$\text{(iv)} \qquad g(\mathbf{w}'/\mathbf{c}, \mathbf{\Pi}^*) - g(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*) = \sum_{t=1}^{T} (f(w'_t) - f(w_t))\,\mathcal{A}(\mathbf{X}, \mathbf{c}, \mathbf{\Pi}^*, t)$$

A Taylor expansion yields:

$$f(w'_t) = f(w_t) + (w'_t - w_t)f'(w_t) + \sum_{k=2}^{\infty} \frac{(w'_t - w_t)^k}{k!} f^{(k)}(w_t)$$

Thus, if

$$\sum_{k=2}^{\infty} \frac{(w'_t - w_t)^k}{k!} f^{(k)}(w_t) \geq 0$$

then

$$f(w'_t) - f(w_t) \geq (w'_t - w_t)f'(w_t)$$

and:

$$\sum_{t=1}^{T}(f(w'_t) - f(w_t))\mathcal{A}(\mathbf{X}, \mathbf{c}, \mathbf{\Pi}^*, t) \geq \sum_{t=1}^{T} f'(w_t)(w'_t - w_t)\mathcal{A}(\mathbf{X}, \mathbf{c}, \mathbf{\Pi}^*, t)$$

that is:

$$g(\mathbf{w}'/\mathbf{c}, \mathbf{\Pi}^*) - g(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*) \geq \nabla g(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*)(\mathbf{w}' - \mathbf{w})$$

Thus, if $\nabla g(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*)(\mathbf{w}' - \mathbf{w}) \geq 0$, then $g(\mathbf{w}'/\mathbf{c}, \mathbf{\Pi}^*) \geq g(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*)$, which proves the implication from Condition 1.□

Proof: $g(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*)$ is pseudo-convex *iff*:

$$\forall\, \mathbf{c}, \mathbf{c}' \;\; \nabla g(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*)\,(\mathbf{c}' - \mathbf{c}) \geq 0 \implies g(\mathbf{c}'/\mathbf{w}, \mathbf{\Pi}^*) \geq g(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*)$$

Let

$$\mathcal{B}(\mathbf{X}, \mathbf{w}, \mathbf{\Pi}^*, t) = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t', t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)$$

where $\mathbf{\Pi}^*$ is the set of all known (and fixed) alignments for the time series $\mathbf{x}$ in $\mathbf{X}$. As only positive quantities are involved, $\mathcal{B}(\mathbf{X}, \mathbf{w}, \mathbf{\Pi}^*, t) \geq 0$. We have:

$$\text{(i)} \quad g(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*) = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t', t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\varphi(x_{t'}, c_t),$$

$$\text{(ii)} \quad \frac{\partial g(c_t/\mathbf{w}, \mathbf{\Pi}^*)}{\partial_{c_t}} = \frac{\partial \varphi(x_{t'}, c_t)}{\partial c_t}\, \mathcal{B}(\mathbf{X}, \mathbf{w}, \mathbf{\Pi}^*, t),$$

$$\text{(iii)} \quad \nabla g(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*)(\mathbf{c}' - \mathbf{c}) = \sum_{t=1}^{T} \frac{\partial \varphi(x_{t'}, c_t)}{\partial c_t}(c'_t - c_t)\, \mathcal{B}(\mathbf{X}, \mathbf{w}, \mathbf{\Pi}^*, t),$$

$$\text{(iv)} \quad g(\mathbf{c}'/\mathbf{w}, \mathbf{\Pi}^*) - g(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*) = \sum_{t=1}^{T} (\varphi(x_{t'}, c'_t) - \varphi(x_{t'}, c_t))\, \mathcal{B}(\mathbf{X}, \mathbf{w}, \mathbf{\Pi}^*, t)$$

A Taylor expansion yields:

$$\varphi(x_{t'}, c'_t) = \varphi(x_{t'}, c_t) + (c'_t - c_t)\frac{\partial \varphi(x_{t'}, c_t)}{\partial c_t} + \sum_{k=2}^{\infty} \frac{(c'_t - c_t)^k}{k!} \frac{\partial^k \varphi(x_{t'}, c_t)}{\partial^k c_t}$$

Thus, if

$$\sum_{k=2}^{\infty} \frac{(c'_t - c_t)^k}{k!} \frac{\partial^k \varphi(x_{t'}, c_t)}{\partial^k c_t} \geq 0$$

then

$$\varphi(x_{t'}, c'_t) - \varphi(x_{t'}, c_t) \geq (c'_t - c_t)\frac{\partial \varphi(x_{t'}, c_t)}{\partial c_t}$$

and:

$$\sum_{t=1}^{T} (\varphi(x_{t'}, c'_t) - \varphi(x_{t'}, c_t))\mathcal{B}(\mathbf{X}, \mathbf{w}, \mathbf{\Pi}^*, t) \geq \sum_{t=1}^{T} (c'_t - c_t)\frac{\partial \varphi(x_{t'}, c_t)}{\partial c_t}\mathcal{B}(\mathbf{X}, \mathbf{w}, \mathbf{\Pi}^*, t)$$

that is:

$$g(\mathbf{c}'/\mathbf{w}, \mathbf{\Pi}^*) - g(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*) \geq \nabla g(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*)(\mathbf{c}' - \mathbf{c})$$

Thus, if $\nabla g(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*)(\mathbf{c}' - \mathbf{c}) \geq 0$, then $g(\mathbf{c}'/\mathbf{w}, \mathbf{\Pi}^*) \geq g(\mathbf{c}/\mathbf{w}, \mathbf{\Pi}^*)$, which proves the implication from Condition 2.□

The proof for Theorem 3.2 directly parallels the ones above, the only difference being in the sign of the inequalities manipulated.

Appendix B

Proof of Solutions 3.15 and 3.16

We present here the proofs for proposed WDTW solutions 3.15 and 3.16. Let us first recall that:

$$g(\mathbf{c}, \mathbf{w}) = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\, \varphi(x_{t'}, c_t)$$

with:

$$f(x) = \begin{cases} x^{-\alpha} & \alpha > 0 \\ e^{-\alpha x} & \alpha > 0 \end{cases}$$

and φ the Euclidean distance.

Given $\mathbf{w} = (w_1, ..., w_T)$ and $\boldsymbol{\Pi}^* = \{\boldsymbol{\pi}^*_{\mathbf{x}} \,/\, \mathbf{x} \in \mathbf{X}\}$, the function defined in Eq. 3.14 is convex in $\mathbf{c}$ and the centroid $\mathbf{c}$ that minimizes the sum of cluster dissimilarities is obtained by solving the partial derivative equation, $\forall t$, $1 \leq t \leq T$:

$$c_t = \frac{\sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} x_{t'}}{\sum_{\mathbf{x} \in \mathbf{X}} \frac{|N(t, \mathbf{x})|}{|\boldsymbol{\pi}^*_{\mathbf{x}}|}} \tag{B.1}$$

where $\boldsymbol{\pi}^*_{\mathbf{x}}$ denotes the optimal alignment for $\mathbf{x} \in \mathbf{X}$ and $|N(t, \mathbf{x})| = \{t' \,/\, (t', t) \in \boldsymbol{\pi}^*_{\mathbf{x}}\}$ denotes the number of time instants of $\mathbf{x}$ aligned to time t of $\mathbf{c}$.

Proof: As φ satisfies Condition 2 of Theorem 3.1, the Karush-Kuhn-Tucker conditions are satisfied and the solution to the minimization problem above is obtained by solving:

$$\frac{\partial g(\mathbf{c}/\mathbf{w}, \boldsymbol{\Pi}^*)}{\partial c} = 0$$

Thus

$$\frac{\partial g(\mathbf{c}/\mathbf{w}, \boldsymbol{\Pi}^*)}{\partial c} = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} 2\, f(w_t)\, (c_t - x_{t'}) = 0$$

then

$$\sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} (c_t - x_{t'}) = 0$$

which leads to:

$$\sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} c_t = \sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} x_{t'}$$

and to Eq.B.1. □

The solution for weight estimation of WDTW is obtained by equating the partial derivative of the Lagrangian of $g(\mathbf{w}/\mathbf{c}, \boldsymbol{\Pi}^*)$, subject to $\sum_{t=1}^{T} w_t = 1$ and $w_t > 0,\ \forall t$, with respect to $\mathbf{w}$ to 0 and solving for $\mathbf{w}$, $\forall t,\ 1 \leq t \leq T$:

for $f(x) = x^{-\alpha}$;

$$w_t = \frac{A_t^{\frac{1}{1+\alpha}}}{\sum_{t=1}^{T} A_t^{\frac{1}{1+\alpha}}} \tag{B.2}$$

for $f(x) = e^{-\alpha x}$;

$$w_t = \frac{1}{\alpha} log \left(\frac{A_t}{(\prod_{t=1}^{T} A_t)^{1/T}} \right) + \frac{1}{T} \tag{B.3}$$

with

$$A_t = \sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} (x_{t'} - c_t)^2$$

Proof: As f satisfies Condition 1 of Theorem 3.1 and the constraints are pseudo-convex, the Karush-Kuhn-Tucker conditions are satisfied. The Lagrangian of the above problem is defined as:

$$L = \sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t,t')\in\boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\ \ \varphi(x_{t'}, c_t) + \lambda(1 - \sum_{t=1}^{T} w_t)$$

and the solution to the above problem is obtained by solving:

$$\frac{\partial L}{\partial w_t} = 0$$

For $f(x) = x^{-\alpha}$ with $\alpha > 0$, one gets:

$$\frac{\partial L}{\partial w_t} = \sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} -\alpha\, w_t^{(-\alpha-1)}\, \varphi(x_{t'}, c_t) - \lambda = 0$$

then

$$\sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} -\alpha\, w_t^{-\alpha-1}\, \varphi(x_{t'}, c_t) = \lambda$$

which leads to:

$$w_t = \frac{\alpha^{1/(\alpha+1)}}{(-\lambda)^{1/\alpha+1}} (A_t)^{1/(\alpha+1)}$$

Summing over t and equating to 1 (the constraint) yields:

$$\sum_{t=1}^{T} w_t = \sum_{t=1}^{T} \frac{\alpha^{1/(\alpha+1)}}{(-\lambda)^{1/\alpha+1}} (A_t)^{1/(\alpha+1)} = 1$$

so,

$$\sum_{t=1}^{T} (A_t)^{1/(\alpha+1)} \cdot \frac{\alpha^{1/(\alpha+1)}}{(-\lambda)^{1/\alpha+1}} = 1$$

thus

$$\lambda = -\alpha\, (\sum_{t=1}^{T} (A_t)^{1/(\alpha+1)})^{\alpha+1}$$

and finally:

$$w_t = \frac{A_t^{\frac{1}{1+\alpha}}}{\sum_{t=1}^{T} A_t^{\frac{1}{1+\alpha}}}$$

.□

For the case $f(x) = e^{-\alpha x}$;

$$\frac{\partial L}{\partial w_t} = \sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} -\alpha\, e^{(-\alpha w_t)}\, \varphi(x_{t'}, c_t) - \lambda = 0$$

then

$$e^{(-\alpha w_t)} = \frac{-\lambda}{\alpha\, A_t}$$

which leads to:

$$w_t = \frac{1}{-\alpha} \log\, (\frac{-\lambda}{\alpha\, A_t}) \tag{B.4}$$

subject to $\sum_{t=1}^{T} w_t = 1$:

$$\sum_{t=1}^{T} \frac{1}{-\alpha} \log\, (\frac{\lambda}{-\alpha\, A_t}) = 1$$

$$\Rightarrow \frac{1}{-\alpha}\sum_{t=1}^{T}\log\left(\frac{\lambda}{-\alpha\, A_t}\right) = 1$$

$$\Rightarrow \frac{1}{-\alpha}\left[\sum_{t=1}^{T}\log\lambda - \log\left(-\alpha\, A_t\right)\right] = 1$$

$$\Rightarrow T.\log\lambda - \sum_{t=1}^{T}\log\left(-\alpha\, A_t\right) = -\alpha$$

$$\Rightarrow T.\log\lambda = \sum_{t=1}^{T}\log\left(-\alpha\, A_t\right) - \alpha$$

$$\Rightarrow T.\log\lambda = T.\log\left(-\alpha\right) + \sum_{t=1}^{T}\log A_t - \alpha$$

thus

$$\log\lambda = \log\left(-\alpha\right) + \frac{\sum_{t=1}^{T}\log A_t}{T} - \frac{\alpha}{T} \tag{B.5}$$

From Eq. B.4, we have:

$$w_t = \frac{1}{-\alpha}\left[\log\lambda - \log\left(-\alpha\, A_t\right)\right]$$

and considering Eq. B.5:

$$w_t = \frac{1}{-\alpha}\left[\left(\log\left(-\alpha\right) + \frac{1}{T}\sum_{t=1}^{T}\log A_t - \frac{\alpha}{T}\right) - \left(\log\left(-\alpha\right) + \log A_t\right)\right]$$

$$\Rightarrow w_t = \frac{1}{-\alpha}\left[\frac{1}{T}\sum_{t=1}^{T}\log A_t - \frac{\alpha}{T} - \log A_t\right]$$

$$= \frac{1}{\alpha}\left[\log A_t - \frac{1}{T}\sum_{t=1}^{T}\log A_t\right] + \frac{1}{T}$$

and finally

$$w_t = \frac{1}{\alpha} log\left(\frac{A_t}{(\prod_{t=1}^{T} A_t)^{1/T}}\right) + \frac{1}{T}$$

Note that, the weight $w_t > 0$ of the element t of $\mathbf{c}$ that ensures the inertia minimization is defined as the proportion of the divergence induced by the elements t' of the series $\mathbf{x} \in \mathbf{X}$ aligned to t. $\square$

Appendix C

Proof of Solutions 3.18 and 3.19

We present here the proof for proposed solutions 3.18 and 3.19. Let us first recall that:

$$g_\kappa(\mathbf{c}, \mathbf{w}) = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\, \kappa(x_{it'}, c_t)$$

with:

$$f(x) = \begin{cases} x^\alpha & \alpha < 1 \\ \log(\alpha x) & \alpha > 0 \end{cases}$$

and κ the Gaussian kernel:

$$\kappa(x, c) = \mathrm{e}^{(-\frac{\|\mathrm{x}-\mathrm{c}\|^2}{2\sigma^2})}$$

(x and c are two real numbers).
Given $\mathbf{w} = (w_1, ..., w_T)$ and $\boldsymbol{\Pi}^* = \{\boldsymbol{\pi}^*_{\mathbf{x}} / \mathbf{x} \in \mathbf{X}\}$, for the centroid $\mathbf{c}$:

$$c_t^{(p+1)} = c_t^{(p)} + \eta^{(p)} \frac{\partial L}{\partial c_t^{(p)}} \text{ and } \eta^{(p+1)} = \frac{\eta^{(p)}}{p} (\eta^{(0)} = 1)$$

with

$$\frac{\partial L}{\partial c_t} = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\, \frac{(x_{t'} - c_t)}{\sigma^2}\, \mathrm{e}^{(-\frac{(\mathrm{x_{t'}}-\mathrm{c_t})^2}{2\sigma^2})} \tag{C.1}$$

Proof:

$$g_k(\mathbf{c}/\mathbf{w}, \boldsymbol{\Pi}^*) = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\, \mathrm{e}^{(-\frac{(\mathrm{x_{t'}}-\mathrm{c_t})^2}{2\sigma^2})}$$

$$\Rightarrow \frac{\partial g_k(\mathbf{c}/\mathbf{w}, \boldsymbol{\Pi}^*)}{\partial c_t} = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\, \frac{2(x_{t'} - c_t)}{2\sigma^2}\, \mathrm{e}^{(-\frac{(\mathrm{x_{t'}}-\mathrm{c_t})^2}{2\sigma^2})}$$

So,

$$\frac{\partial g_k(\mathbf{c}/\mathbf{w}, \boldsymbol{\Pi}^*)}{\partial c_t} = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\, \frac{(x_{t'} - c_t)}{\sigma^2}\, \mathrm{e}^{(-\frac{(\mathrm{x_{t'}}-\mathrm{c_t})^2}{2\sigma^2})}.\square$$

Given $\mathbf{c} = (c_1, ..., c_T)$ and $\Pi^* = \{\boldsymbol{\pi}^*_{\mathbf{x}} / \mathbf{x} \in \mathbf{X}\}$, the weight vector $\mathbf{w}$ that maximizes the sum of intra-cluster similarities $g_\kappa(\mathbf{c}, \mathbf{w}, \boldsymbol{\Pi}^*)$, subject to $\sum_{t=1}^{T} w_t = 1$ and $w_t > 0,\ \forall t$, is defined by:

for $f(x) = x^\alpha$;

$$w_t = \frac{A_t^{\frac{1}{1-\alpha}}}{\sum_{t=1}^{T} A_t^{\frac{1}{1-\alpha}}} \tag{C.2}$$

for $f(x) = \log(\alpha x)$;

$$w_t = \frac{A_t}{\sum_{t=1}^{T} A_t} \tag{C.3}$$

with

$$A_t = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} e^{(-\frac{(x_{t'} - c_t)^2}{2\sigma^2})}$$

Proof: As f satisfies Condition 3 of Theorem 3.2 and the constraints are pseudo-concave, the Karush-Kuhn-Tucker conditions are satisfied. The Lagrangian of the above problem is defined as:

$$L = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t,t') \in \boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\ \ \kappa(x_{it'}, c_t) + \lambda(1 - \sum_{t=1}^{T} w_t)$$

and the solution to the above problem is obtained by solving $\frac{\partial L}{\partial w} = 0$.

For $f(x) = x^\alpha$ $(0 \leq \alpha \leq 1)$:

$$\frac{\partial L}{\partial w_t} = \alpha w_t^{(\alpha-1)} A_t - \lambda = 0$$

$$\Rightarrow w_t^{(\alpha-1)} = \frac{\lambda}{\alpha A_t}$$

and

$$w_t = \frac{\alpha^{\frac{1}{1-\alpha}} A_t^{\frac{1}{1-\alpha}}}{\lambda^{\frac{1}{1-\alpha}}} \tag{C.4}$$

Summing over t and equating to 1 (the constraint) leads to:

$$\sum_{t=1}^{T} \frac{\alpha^{\frac{1}{1-\alpha}} A_t^{\frac{1}{1-\alpha}}}{\lambda^{\frac{1}{1-\alpha}}} = 1$$

$$\Rightarrow \frac{\alpha^{\frac{1}{1-\alpha}}}{\lambda^{\frac{1}{1-\alpha}}} \sum_{t=1}^{T} A_t^{\frac{1}{1-\alpha}} = 1$$

and

$$\lambda^{(\frac{1}{1-\alpha})} = \alpha^{(\frac{1}{1-\alpha})} \sum_{t=1}^{T} A_t^{(\frac{1}{1-\alpha})} \tag{C.5}$$

From Eq. C.4 and replacing the value for λ in the previous Eq. C.5, we have:

$$w_t = \frac{\alpha^{\frac{1}{1-\alpha}} A_t^{\frac{1}{1-\alpha}}}{\alpha^{(\frac{1}{1-\alpha})} \sum_{t=1}^{T} A_t^{(\frac{1}{1-\alpha})}}$$

and finally yields Eq. C.2:

$$w_t = \frac{A_t^{\frac{1}{1-\alpha}}}{\sum_{t=1}^{T} A_t^{\frac{1}{1-\alpha}}}$$

.□

For $f(x) = \log(\alpha x)$ $(\alpha > 0)$:

$$\frac{\partial L}{\partial w_t} = \frac{1}{w_t} A_t - \lambda = 0$$

so,

$$w_t = \frac{A_t}{\lambda} \tag{C.6}$$

subject to $\sum_{t=1}^{T} w_t = 1$:

$$\sum_{t=1}^{T} \frac{A_t}{\lambda} = 1$$

and

$$\lambda = \sum_{t=1}^{T} A_t \tag{C.7}$$

From Eq. C.6 and C.7, one can easily obtain Eq. C.3:

$$w_t = \frac{A_t}{\sum_{t=1}^{T} A_t}$$

.□

Appendix D

Proof of Solutions 3.21 and 3.22

Let us first recall that:

$$g_\kappa(\mathbf{c}, \mathbf{w}) = \sum_{i=1}^{N} \mathrm{WK}_{\mathrm{GDTW}}(\mathbf{x}_i, (\mathbf{c}, \mathbf{w}))$$

$$= \sum_{\mathbf{x} \in \mathbf{X}} \underbrace{exp\left[\frac{-1}{\lambda} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\ (x_{t'} - c_t)^2 \right]}_{A(\mathbf{w},\mathbf{c})}$$

with:

$$f(x) = \begin{cases} x^{-\alpha} & \alpha > 0 \\ e^{-\alpha x} & \alpha > 0 \end{cases}$$

Given $\mathbf{w} = (w_1, ..., w_T)$ and $\boldsymbol{\Pi}^* = \{\boldsymbol{\pi}^*_{\mathbf{x}} \,/\, \mathbf{x} \in \mathbf{X}\}$, for the centroid $\mathbf{c}$:

$$c_t^{(p+1)} = c_t^{(p)} + \eta^{(p)}\ \frac{\partial L}{\partial c_t^{(p)}} \text{ and } \eta^{(p+1)} = \frac{\eta^{(p)}}{p} (\eta^{(0)} = 1)$$

with

$$\frac{\partial L}{\partial c_t} = \frac{-2}{\lambda} \left[\sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \left(\sum_{t'' \in T, (t'',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)(c_t - x_{t''}) \right) .\ A(\mathbf{w}, \mathbf{c}) \right] \tag{D.1}$$

Proof:

$$g_k(\mathbf{c}/\mathbf{w}, \boldsymbol{\Pi}^*) = \sum_{\mathbf{x} \in \mathbf{X}} exp\left[\frac{-1}{\lambda} \underbrace{\frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t) \in \boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\ (x_{t'} - c_t)^2}_{F(\mathbf{c})} \right]$$

$$= \sum_{\mathbf{x} \in \mathbf{X}} exp\left(\frac{-1}{\lambda}\ F(\mathbf{c}) \right)$$

$$\Rightarrow \frac{\partial g_k(\mathbf{c}/\mathbf{w}, \boldsymbol{\Pi}^*)}{\partial c_t} = \sum_{\mathbf{x} \in \mathbf{X}} \frac{-1}{\lambda}\ \frac{\partial F(\mathbf{c})}{\partial c_t}\ .\ exp\left(\frac{-1}{\lambda}\ F(\mathbf{c}) \right)$$

$$= \sum_{\mathbf{x}\in\mathbf{X}} \frac{-1}{\lambda\ |\boldsymbol{\pi}^*_{\mathbf{x}}|} \left[\sum_{t''\in T,(t'',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\ 2\ (c_t - x_{t''}) \right] .exp\left(\frac{-1}{\lambda}\ F(\mathbf{c})\right)$$

So,:

$$\frac{\partial g_k(\mathbf{c}/\mathbf{w},\boldsymbol{\Pi}^*)}{\partial c_t} = \frac{-2}{\lambda} \left[\sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \left(\sum_{t''\in T,(t'',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\ (c_t - x_{t''}) \right) .\ A(\mathbf{w},\mathbf{c}) \right]$$

.□

The solution for the weights vector $\mathbf{w}$, is obtained by equating the partial derivative of the Lagrangian of $g_\kappa(\mathbf{c}/\mathbf{w},\boldsymbol{\Pi}^*)$ with respect to $\mathbf{w}$ to 0 and solving for $\mathbf{w}$:

$$w_t^{(p+1)} = w_t^{(p)} + \eta^{(p)}\ \frac{\partial L_w}{\partial c_t^{(p)}} \text{ and } \eta^{(p+1)} = \frac{\eta^{(p)}}{p}(\eta^{(0)} = 1)$$

with

$$\frac{\partial L_w}{\partial w_t} = \frac{\alpha}{\lambda}\ w_t^{-(\alpha+1)} \left[\sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \left(\sum_{t''\in T,(t'',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} (x_{t''} - c_t)^2 \right) .\ A(\mathbf{w},\mathbf{c}) \right] \tag{D.2}$$

for $f(x) = x^{-\alpha}$, and

$$\frac{\partial L_w}{\partial w_t} = \frac{\alpha}{\lambda}\ e^{(-\alpha w_t)} \left[\sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \left(\sum_{t''\in T,(t'',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} (x_{t''} - c_t)^2 \right) .\ A(\mathbf{w},\mathbf{c}) \right] \tag{D.3}$$

for $f(x) = e^{-\alpha x}$.

Proof:

$$g_k(\mathbf{w}/\mathbf{c},\boldsymbol{\Pi}^*) = \sum_{\mathbf{x}\in\mathbf{X}} \underbrace{exp\left[\frac{-1}{\lambda}\ \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \sum_{(t',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} f(w_t)\ (x_{t'} - c_t)^2 \right]}_{A(\mathbf{w},\mathbf{c})}$$

for $f(x) = x^{-\alpha}$:

$$\frac{\partial g_k(\mathbf{w}/\mathbf{c},\boldsymbol{\Pi}^*)}{\partial w_t} = \frac{\alpha}{\lambda}\ w_t^{-(\alpha+1)} \left[\sum_{\mathbf{x}\in\mathbf{X}} \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|} \left(\sum_{t''\in T,(t'',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} (x_{t''} - c_t)^2 \right) .\ A(\mathbf{w},\mathbf{c}) \right]$$

and for $f(x) = e^{-\alpha x}$:

$$\frac{\partial g_k(\mathbf{w}/\mathbf{c}, \mathbf{\Pi}^*)}{\partial w_t} = \frac{\alpha}{\lambda}\ e^{(-\alpha w_t)}\ \left[\sum_{\mathbf{x}\in\mathbf{X}}\ \frac{1}{|\boldsymbol{\pi}^*_{\mathbf{x}}|}\ \left(\sum_{t''\in T,(t'',t)\in\boldsymbol{\pi}^*_{\mathbf{x}}} (x_{t''} - c_t)^2\right).\ A(\mathbf{w}, \mathbf{c})\right]$$

.□

Appendix E

Proofs for the Case of $\mathrm{WK}_{\mathrm{GA}}$

Let us first recall Theorem 3.2.

Theorem 3.2 *Let $f : [0,1] \to \mathbb{R}^+$ be the non-decreasing function used in g_κ, let κ be the positive definite symmetric kernel used in g_κ and let Conditions 3 and 4 be defined as:*

$$\textbf{Condition 3:} \quad \forall a \in [-1,1],\ \forall x \in [0,1],\ \sum_{k=2}^{\infty} \frac{a^k}{k!} f^{(k)}(x) \leq 0$$

$$\textbf{Condition 4:} \quad \forall (c,c') \in \mathbb{R}^2, \forall x \in \mathbb{R}, \sum_{k=2}^{\infty} \frac{(c'-c)^k}{k!} \frac{\partial^k \kappa(x,c)}{\partial c^k} \leq 0$$

Then:

If Condition 3 holds, then $g_\kappa(\boldsymbol{w}/\boldsymbol{c}, \boldsymbol{\Pi}^*)$ is pseudo-concave;
If Condition 4 holds, then $g_\kappa(\boldsymbol{c}/\boldsymbol{w}, \boldsymbol{\Pi}^*)$ is pseudo-concave.

Proof: $g(\mathbf{w}/\mathbf{c}, \boldsymbol{\Pi}^*)$ is pseudo-concave *iff*:

$$\forall\, \mathbf{w}, \mathbf{w}' \ \ \nabla g(\mathbf{w}/\mathbf{c}, \boldsymbol{\Pi}^*)\,(\mathbf{w}' - \mathbf{w}) \leq 0 \implies g(\mathbf{w}'/\mathbf{c}, \boldsymbol{\Pi}^*) \leq g(\mathbf{w}/\mathbf{c}, \boldsymbol{\Pi}^*)$$

Let

$$g_\kappa(\mathbf{c}, \mathbf{w}) = \sum_{\mathbf{x} \in \mathbf{X}} \mathrm{WK}_{\mathrm{GA}}(\mathbf{x}, (\mathbf{c}, \mathbf{w}))$$

$$= \sum_{\mathbf{x} \in \mathbf{X}} \left[\sum_{\boldsymbol{\pi} \in \mathbb{A}} \prod_{(t',t) \in \boldsymbol{\pi}} f(w_t)\ k(x_{t'}, c_t) \right]$$

We have:

(**i**) $$g_k(\mathbf{w}, \mathbf{c}) = \sum_{\mathbf{x} \in \mathbf{X}} \left[\sum_{\boldsymbol{\pi} \in \mathbb{A}} \prod_{(t',t) \in \boldsymbol{\pi}} f(w_t)\ k(x_{t'}, c_t) \right]$$

(**ii**) $$\frac{\partial g_k(w_t, \mathbf{c})}{\partial w_t} = \sum_{\mathbf{x} \in \mathbf{X}} \sum_{\boldsymbol{\pi} \in \mathbb{A}} g_k(\mathbf{c}, \mathbf{w}, \mathbf{X}, \boldsymbol{\pi})\ .\ n(\boldsymbol{\pi}, t)\ \frac{f'(w_t)}{f(w_t)}$$

$$= \frac{f'(w_t)}{f(w_t)} \sum_{\mathbf{x}\in\mathbf{X}} \sum_{\boldsymbol{\pi}\in\mathbb{A}} g_k(\mathbf{c},\mathbf{w},\mathbf{X},\boldsymbol{\pi}) \;.\; n(\boldsymbol{\pi},t)$$

(iii) $$\nabla g_k(\mathbf{w}).(w'-w) = \sum_{t=1}^{T} \frac{f'(w_t)}{f(w_t)} \; (w'_t - w_t) \sum_{\mathbf{x}\in\mathbf{X}} \sum_{\boldsymbol{\pi}\in\mathbb{A}} g_k(\mathbf{c},\mathbf{w},\mathbf{X},\boldsymbol{\pi}) \;.\; n(\boldsymbol{\pi},t)$$

(iv) $$g_k(\mathbf{w}'|\mathbf{c}) - g_k(\mathbf{w}|\mathbf{c})$$

$$= \sum_{\mathbf{x}\in\mathbf{X}} \left[\sum_{\boldsymbol{\pi}\in\mathbb{A}} \prod_{(t',t)\in\boldsymbol{\pi}} f(w'_t)\; k(x_{t'},c_t) \right] - \sum_{\mathbf{x}\in\mathbf{X}} \left[\sum_{\boldsymbol{\pi}\in\mathbb{A}} \prod_{(t',t)\in\boldsymbol{\pi}} f(w_t)\; k(x_{t'},c_t) \right]$$

$$= \underbrace{\sum_{\mathbf{x}\in\mathbf{X}} \sum_{\boldsymbol{\pi}\in\mathbb{A}} \left[\prod_{(t',t)\in\boldsymbol{\pi}} f(w'_t)\; k(x_{t'},c_t) - \prod_{(t',t)\in\boldsymbol{\pi}} f(w_t)\; k(x_{t'},c_t) \right]}_{\xi}$$

where,

$$\prod_{(t',t)\in\boldsymbol{\pi}} f(w_t)\; k(x_{t'},c_t) \;=\; \prod_{t=1}^{T} \prod_{t'\in\boldsymbol{\pi}(t)} f(w_t)\; k(x_{t'},c_t) \;=\; \prod_{t=1}^{T} f(w_t)^{n(\boldsymbol{\pi},t)} \prod_{t'\in\boldsymbol{\pi}(t)} k(x_{t'},c_t)$$

Thus,

$$\xi = \sum_{\mathbf{x}\in\mathbf{X}} \sum_{\boldsymbol{\pi}\in\mathbb{A}} \left[\prod_{t=1}^{T} f(w'_t)^{n(\boldsymbol{\pi},t)} \prod_{t'\in\boldsymbol{\pi}(t)} k(x_{t'},c_t) \;-\; \prod_{t=1}^{T} f(w_t)^{n(\boldsymbol{\pi},t)} \prod_{t'\in\boldsymbol{\pi}(t)} k(x_{t'},c_t) \right]$$

$$= \sum_{\mathbf{x}\in\mathbf{X}} \sum_{\boldsymbol{\pi}\in\mathbb{A}} \left(\prod_{t=1}^{T} \prod_{t'\in\boldsymbol{\pi}(t)} k(x_{t'},c_t) \right) \;.\; \left[\prod_{t=1}^{T} f(w'_t)^{n(\boldsymbol{\pi},t)} \;-\; \prod_{t=1}^{T} f(w_t)^{n(\boldsymbol{\pi},t)} \right]$$

Suppose;

$$A_1 = \sum_{t=1}^{T} \frac{f'(w_t)}{f(w_t)} \;.\; (w'_t - w_t) \sum_{\mathbf{x}\in\mathbf{X}} \sum_{\boldsymbol{\pi}\in\mathbb{A}} \prod_{(t_1,t_2)\in\boldsymbol{\pi}} f(w_{t_2}) \;.\; n(\boldsymbol{\pi},t) \;.\; k(x_{t_1},c_{t_2})$$

and

$$A_2 = \sum_{\mathbf{x}\in\mathbf{X}} \sum_{\boldsymbol{\pi}\in\mathbb{A}} \prod_{(t_1,t_2)\in\boldsymbol{\pi}} k(x_{t_1}, c_{t_2}) \; . \; \left[\prod_{t=1}^{T} f(w'_t)^{n(\boldsymbol{\pi},t)} \; - \; \prod_{t=1}^{T} f(w_t)^{n(\boldsymbol{\pi},t)} \right]$$

We can not conclude if $A_1 \leq 0 \;\; \Rightarrow \;\; A_2 \leq 0$. Therefore, there is no proof for the pseudo-concavity of $g_k(\mathbf{w}, \mathbf{c})$ and we can not benefit from the recurrence formulas of $\mathrm{WK}_{\mathrm{GA}}$.$\square$

Bibliography

[AC01] J. Aach and G. M. Church. "Aligning gene expression time series with time warping algorithms." In: *Bioinformatics*. Vol. 17. 6. 2001, 495—508 (cit. on p. 32).

[ACS03] W.H. Abdulla, D. Chow, and G. Sin. "Cross-words reference template for DTW-based speech recognition sys- tems." In: *Proc. TENCON*. Vol. 2. 2003, pp. 1576–1579 (cit. on pp. 35, 47).

[AT10] Z. Abraham and P. Tan. "An integrated framework for simultaneous classification and regression of time-series Data." In: *SIAM International Conference on Data Mining*. 2010, pp. 653–664 (cit. on p. 18).

[AV07] David Arthur and Sergei Vassilvitskii. "K-means++: The Advantages of Careful Seeding." In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (2007), pp. 1027–1035 (cit. on pp. 2, 41, 47, 49).

[Bai12] Werner Bailer. *Sequence kernels for clustering and visualizing near duplicate video segments*. Springer, 2012 (cit. on p. 50).

[BF98] Paul S. Bradley and Usama M. Fayyad. "Refining Initial Points for K-Means Clustering." In: *Proceedings of the Fifteenth International Conference on Machine Learning*. ICML '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 91–99 (cit. on pp. 1, 47).

[BH65] G. Ball and D. Hall. "ISODATA: A novel method of data analysis and pattern classification." In: *Stanford Research Institute* (1965) (cit. on p. 41).

[BHB02] Claus Bahlmann, Bernard Haasdonk, and Hans Burkhardt. "Online handwriting recognition with support vector machines-a kernel approach." In: *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*. IEEE. 2002, pp. 49–54 (cit. on pp. 2, 24, 47, 48, 50, 52).

[BW95] A. Bruderlin and L. Williams. "Motion signal processing." In: *ACM SIGGRAPH*. 1995, 97—104 (cit. on p. 32).

[Cab+07] F. Cabestaing et al. "Classification of evoked potentials by Pearsonís correlation in a Brain-Computer Interface." In: *Modelling C Automatic Control (theory and applications)* 67 (2007), pp. 156–166 (cit. on p. 18).

[CC01] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Chapman and Hall., 2001 (cit. on p. 66).

[CI02] Y. Caspi and M. Irani. "Aligning non-overlapping sequences." In: *Int'l J. Computer Vision*. Vol. 48. 1. 2002, 39—51 (cit. on p. 32).

[CL71] William W Cooley and Paul R Lohnes. *Multivariate data analysis*. J. Wiley, 1971 (cit. on p. 66).

[CL88] Humberto Carrillo and David Lipman. "The multiple sequence alignment problem in biology." In: *SIAM Journal on Applied Mathematics* 48.5 (1988), pp. 1073–1082 (cit. on pp. 30, 33).

[CNH00] D.J. Higgins Cédric Notredame and Jaap Heringa. "T-Coffee: A novel method for fast and accurate multiple sequence alignment." In: *Journal of molecular biology* 302.1 (2000), pp. 205–217 (cit. on pp. 2, 30, 33, 47).

[Cut+07] M. Cuturi et al. "A kernel for time series based on global alignments." In: *the International Conference on Acoustics, Speech and Signal Processing*. Vol. 11. 2007, pp. 413–416 (cit. on pp. 2, 25, 47, 48, 50, 53, 58, 62, 68).

[Cut11] Marco Cuturi. "Fast global alignment kernels." In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 2011, pp. 929–936 (cit. on pp. 2, 25, 47, 48, 50, 58, 62, 68).

[DCA12] A. Douzal-Chouakria and C. Amblard. "Classification trees for time series." In: *Pattern Recognition* 45.3 (2012), pp. 1076–1091 (cit. on pp. 19, 21).

[DCN07] Ahlame Douzal-Chouakria and Panduranga Naidu Nagabhushan. "Adaptive dissimilarity index for measuring time series proximity." In: *Advances in Data Analysis and Classification* 1.1 (2007), pp. 5–21 (cit. on p. 21).

[DE93] Banfield J. D. and Raftery A. E. "Model-based Gaussian and non-Gaussian clustering." In: *Biometrics*. Vol. 49. 1993, pp. 803–821 (cit. on p. 40).

[DGK04] Inderjit.S. Dhillon, Yuqiang Guan, and Brian Kulis. "Kernel k-means: spectral clustering and normalized cuts." In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2004, pp. 551–556 (cit. on pp. 2, 47).

[DS02] D. Deoste and B. Scholkopf. "Trainging invariant support vector machines." In: *Machine learning, 46* (2002) (cit. on p. 24).

[DWK05] Doheon Lee Ki-Youngand Hyung Lee Kwang Dae-Won Kim. "Evaluation of the performance of clustering algorithm in kernel-induced feature space." In: *Pattern Recognition* 34.4 (2005), pp. 607–611 (cit. on p. 45).

[ENBJ05] J. Ernst, GJ Nau, and Z. Bar-Joseph. "Clustering short time series gene expression data." In: *Bioinformatics* 21 (2005), pp. 159–168 (cit. on p. 18).

[F.88] Corpet F. "Multiple sequence alignment with hierarchical clustering." In: *Nucleic Acids Research* 16 (1988) (cit. on p. 1).

[FDCG13] C. Frambourg, A. Douzal-Chouakria, and E. Gaussier. "Learning Multiple Temporal Matching for Time Series Classification." In: *Intelligent Data Analysis*. Ed. by A. Tucker et al. London, 2013, pp. 198–209 (cit. on pp. 2, 47, 69).

[FR98] C. Fraley and A.E. Raftery. "How many clusters? Which clustering methods? Answers via model-based cluster analysis." In: *Computer Journal* 41 (1998), pp. 578–588 (cit. on p. 40).

[Fu11] Tak-chung Fu. "A Review on Time Series Data Mining." In: *Eng. Appl. Artif. Intell.* 24.1 (Feb. 2011), pp. 164–181 (cit. on p. 8).

[GDMS96] Lalit Gupta, Ravi Tammana Dennis Molfese, and P.G. Simos. "Nonlinear alignment and averaging for estimating the evoked potential." In: *Biomedical Engineering, IEEE Transactions on* 43.4 (1996), pp. 348–356 (cit. on p. 33).

[GHS11] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. "A time series kernel for action recognition." In: *British Machine Vision Conference*. 2011 (cit. on pp. 20, 23).

[Gir02] Mark Girolami. "Mercer kernel-based clustering in feature space." In: *Neural Networks, IEEE Transactions on* 13.3 (2002), pp. 780–784 (cit. on pp. 2, 47, 66).

[GRS98] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. "CURE: An Efficient Clustering Algorithm for Large Databases." In: *SIGMOD Rec.* 27.2 (June 1998), pp. 73–84 (cit. on p. 1).

[Has+05] Trevor Hastie et al. "The elements of statistical learning: data mining, inference and prediction." In: *The Mathematical Intelligencer* 27.2 (2005), pp. 83–85 (cit. on pp. 48, 49).

[HBV01] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. "On Clustering Validation Techniques." In: *Journal of Intelligent Information Systems* 17 (2001), pp. 107–145 (cit. on p. 1).

[HH08] BOCK Hans-Hermann. "Origins and extensions of the k-means algorithm in cluster analysis." In: *Journal Electronique d'Histoire des Probabilités et de la Statistique Electronic Journal for History of Probability and Statistics* 4 (2008) (cit. on pp. 48, 49).

[HK01] J. Han and M. Kamber. "Data Mining: Concepts and Techniques." In: *Morgan Kaufmann Publishers, USA* (2001) (cit. on p. 40).

[HK02] B. Haasdonk and D. Keysers. "Tangent distance kernels for support vector machines." In: *16th ICPR* (2002) (cit. on p. 24).

[HNF08] V. Hautamaki, P. Nykanen, and P. Franti. "Time-series clustering by approximate prototypes." In: *19th International Conference on Pattern Recognition*. 2008 (cit. on p. 36).

[HR11] Paul Honeine and Cedric Richard. "Preimage problem in kernel-based machine learning." In: *IEEE Signal Processing Magazine* 28.2 (2011), pp. 77–88 (cit. on p. 58).

[HSS08] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. "Kernel methods in machine learning." In: *Annals of Statistics* 36.3 (2008), pp. 1171–1220 (cit. on p. 21).

[Ita75] F. Itakura. "Minimum prediction residual principle applied to speech recognition." In: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 23.1 (1975), pp. 67–72 (cit. on pp. 11, 14, 29, 51).

[Jac93] Donald A Jackson. "Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches." In: *Ecology, JSTOR* (1993), pp. 2204–2214 (cit. on p. 66).

[JER09] C. Joder, S. Essid, and G. Richard. "Temporal integration for audio classification with application to musical instrument classification." In: *IEEE Transactions on Audio, Speech and Language Processing*. 2009, pp. 174–186 (cit. on p. 25).

[JHT01] M. Kamber J. Han and A. Tung. "Spatial clustering methods in data mining: A survey." In: *Geographic Data Mining and Knowledge Discovery, Taylor and Francis* (2001) (cit. on p. 45).

[Jus01] Winfried Just. "Computational complexity of multiple sequence alignment with SP-score." In: *Journal of computational biology* 8.6 (2001), pp. 615–623 (cit. on pp. 30, 33).

[KGP01] Konstantinos Kalpakis, Dhiral Gada, and Vasundhara Puttagunta. "Distance Measures for Effective Clustering of ARIMA Time-Series." In: *Proceedings of the 2001 IEEE International Conference on Data Mining*. ICDM '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 273–280 (cit. on pp. 1, 47).

[KK02] E. Keogh and S. Kasetty. "On the need for time series data mining benchmarks: A survey and empirical." In: *Knowl. Data Discov.* (2002), 102–111 (cit. on p. 2).

[KL83] J.B Kruskall and M. Liberman. *The symmetric time warping algorithm: From continuous to discrete. In Time Warps, String Edits and Macromolecules.* Addison-Wesley., 1983 (cit. on pp. 2, 11, 29, 47, 48, 50, 51).

[KLT03] Eamonn Keogh, Jessica Lin, and Wagner Truppel. "Clustering of Time Series Subsequences is Meaningless: Implications for Previous and Future Research." In: *Proceedings of the Third IEEE International Conference on Data Mining, ICDM '03*. Washington, DC, USA: IEEE Computer Society, 2003, pp. 115– (cit. on pp. 1, 47).

[KP00] E.J. Keogh and M.J. Pazzani. "Scaling Up Dynamic Time Warping for Data Mining Applications." In: *ACM SIGKDD*. 2000, pp. 285–289 (cit. on pp. 15, 16).

[KR87] L. Kaufman and P.J. Rousseeuw. "Clustering by means of Medoids, in Statistical Data Analysis Based on the L1–Norm and Related Methods." In: *North-Holland* (1987), 405–416 (cit. on pp. 2, 43).

[KR90] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data. An Introduction to Cluster Analysis.* John Wiley & Sons, New York., 1990 (cit. on pp. 44, 45).

[Lia05] W.T. Liao. "Clustering of time series data - A survey." In: *Pattern Recognition* 38 (2005), pp. 1857–1874 (cit. on pp. 2, 47).

[Lis+05] J. Listgarten et al. "Multiple alignment of continuous time series." In: *In Proceeding's of the Neural Information Processing Systems*. 2005 (cit. on p. 32).

[LOW02] Weiqiang Lin, Mehmet A. Orgun, and Graham J. Williams. "An Overview of Temporal Data Mining." In: *Proceedings of the 1st Australian Data Mining Workshop*. 2002 (cit. on p. 1).

[LRR04] Brigitte Le Roux and Henry Rouanet. *Geometric data analysis: from correspondence analysis to structured data analysis*. Springer Science & Business Media, 2004 (cit. on p. 72).

[Mac+10] B.D. MacArthur et al. "GATE: Software for the analysis and visualization of high-dimensional time series expression data," in: *Bioinformatics* 26.1 (2010), pp. 143–144 (cit. on p. 18).

[Mac67] J. MacQueen. "Some methods for classification and analysis of multivariate observations." In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967, pp. 281–297 (cit. on pp. 2, 41).

[MG14] Pierre-François Marteau and Sylvie Gibet. "On Recursive Edit Distance Kernels with Application to Time Series Classification." In: *IEEE Transactions on Neural Networks and Learning Systems* 26.6 (2014). 14 pages, pp. 1121–1133 (cit. on p. 47).

[Mil+99] R. T. Miller et al. "A Comprehensive Approach to Clustering of Expressed Human Gene Sequence: The Sequence Tag Alignment and Consensus Knowledge Base." In: *Genome Research* 9 (1999), 1143—1155 (cit. on p. 1).

[MR05] Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005 (cit. on p. 8).

[NH94] R. Ng and J. Han. "Effecient and Effictive Clustering Methods for Spatial Data Mining." In: *In Proceeding's of the 20th VLDB Conference, Santiago, Chile*. 1994 (cit. on p. 45).

[Nom02] Hiroshi Shimodaira Ken-ichi Noma. "Dynamic time-alignment kernel in support vector machine." In: *Advances in neural information processing systems* 14 (2002), p. 921 (cit. on p. 50).

[NR07] Vit Niennattrakul and C.Ann Ratanamahatana. "On clustering multimedia time series data using k-means and dynamic time warping." In: *Multimedia and Ubiquitous Engineering, 2007. MUE'07. International Conference on*. IEEE. 2007, pp. 733–738 (cit. on p. 34).

[NR09] Vit Niennattrakul and C.Ann Ratanamahatana. "Shape averaging under time warping." In: *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th International Conference on*. Vol. 2. IEEE. 2009, pp. 626–629 (cit. on p. 34).

[Osa+02] Naoki Osato et al. "A computer-based method of selecting clones for a full-length cDNA project: Simulataneous collection of negligibly redundant and variant cDNAs." In: *Genome Research* 12 (2002), 1127–1134 (cit. on p. 1).

[Pea96] K. Pearson. "Contributions to the mathematical theory of evolution." In: *Trans. R. Soc. Lond. Ser.* (1896), 253–318 (cit. on p. 18).

[Pir09] Chris Piro. "Chat reaches 1 billion messages sent per day." In: 2009 (cit. on p. 8).

[PKG11] F. Petitjean, A. Ketterlin, and P. GanÇarski. "A global averaging method for dynamic time warping, with applications to clustering." In: *Pattern Recognition* 44.3 (2011), pp. 678–693 (cit. on pp. 36, 37, 47).

[Rab89] L.R. Rabiner. "A Tutorial on Hidden Markov Models and selected applications in speech recognition." In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286 (cit. on p. 29).

[Ran71] William M Rand. "Objective criteria for the evaluation of clustering methods." In: *Journal of the American Statistical association* 66.336 (1971), pp. 846–850 (cit. on p. 68).

[RBK08] J. Rydell, M. Borga, and H. Knutsson. "Robust Correlation Analysis with an Application to Functional MRI." In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 453-456. 2008 (cit. on p. 18).

[RD62] R.Bellman and S. Dreyfus. "Applied Dynamic Programming." In: *New Jersey: Princeton Univ. Press* (1962) (cit. on p. 11).

[RJ93] L. Rabiner and B. Juang. "Fundamentals of speech recognition." In: *Prentice Hall*. 1993 (cit. on p. 32).

[RTZ10] Elisa Ricci, Francesco Tobia, and Gloria Zen. "Learning Pedestrian Trajectories with Kernels." In: *ICPR, IEEE Computer Society*. IEEE Computer Society, 2010, pp. 149–152 (cit. on p. 25).

[SC04] Stan Salvador and Philip Chand. "FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space." In: *KDD Workshop on Mining Temporal and Sequential Data*. 2004, pp. 70–80 (cit. on p. 16).

[SC71] H. Sakoe and S. Chiba. "A dymanic programming approach to continuous speech recognition." In: *Proceedings of the seventh International Congress on Acoustics* 3 (1971), pp. 65–69 (cit. on pp. 14, 51).

[SC78] H. Sakoe and S. Chiba. "Dynamic programming algorithm optimization for spoken word recognition." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26.1 (1978), pp. 43–49 (cit. on pp. 2, 14, 25, 29, 51).

[Shi+02] Hiroshi Shimodaira et al. "Dynamic time-alignment kernel in support vector machine." In: *NIPS*. Vol. 14. 2002, pp. 921–928 (cit. on pp. 2, 24, 47, 48, 50–52, 58).

[SI84] Shokri Z. Selim and M. A. Ismail. "K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality." In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 6.1 (Jan. 1984), pp. 81–87 (cit. on pp. 42, 48, 49).

[SK83] D. Sankoff and J.B. Kruskal. *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison*. Addison-Wesley, 1983 (cit. on pp. 29, 51).

[SKDCG15] Saeid Soheily-Khah, Ahlame Douzal-Chouakria, and Eric Gaussier. "Progressive and Iterative Approaches for Time Series Averaging." In: *ECML-PKDD, Proceedings of AALTD, Porto, Portugal*. 2015 (cit. on p. 37).

[SKDCG16a] Saeid Soheily-Khah, Ahlame Douzal-Chouakria, and Eric Gaussier. "A Comparison of Progressive and Iterative Centroid Estimation Approaches Under Time Warp." In: *Lecture Notes in Computer Science, Advanced Analysis and Learning on Temporal Data* 9785 (2016), pp. 144–156 (cit. on pp. 30, 37).

[SKDCG16b] Saeid Soheily-Khah, Ahlame Douzal-Chouakria, and Eric Gaussier. "Generalized k-means-based clustering for temporal data under weighted and kernel time warp." In: *Journal of Pattern Recognition Letters* (2016) (cit. on pp. 37, 50).

[SLY06] Sing-Hoi Sze, Yue Lu, and Qingwu Yang. "A polynomial time solvable formulation of multiple sequence alignment." In: *Journal of Computational Biology* 13.2 (2006), pp. 309–319 (cit. on pp. 30, 33).

[THG94] Julie D Thompson, Desmond G Higgins, and Toby J Gibson. "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice." In: *Nucleic acids research* 22.22 (1994), pp. 4673–4680 (cit. on pp. 2, 30, 33, 34, 47).

[TK99] S. Theodoridis and K. Koutroubas. In: *Pattern Recognition, Academic Press* (1999) (cit. on p. 1).

[VS10] Gerben de Vries and Maarten van Someren. "Clustering Vessel Trajectories with Alignment Kernels under Trajectory Compression." In: *ECML/PKDD, Lecture Notes in Computer Science*. Vol. 6321. Lecture Notes in Computer Science. Springer, 2010, pp. 296–311 (cit. on p. 25).

[WJ94] Lusheng Wang and Tao Jiang. "On the complexity of multiple sequence alignment." In: *Journal of computational biology* 1.4 (1994), pp. 337–348 (cit. on p. 2).

[WW00] Changzhou Wang and Xiaoyang Sean Wang. "Supporting Content-Based Searches on Time Series via Approximation." In: *Proceedings of the 12th International Conference on Scientific and Statistical Database Management*. SSDBM '00. Washington, DC, USA: IEEE Computer Society, 2000, pp. 69– (cit. on p. 47).

[Zho+05] Wei Zhong et al. "Improved K-Means Clustering Algorithm for Exploring Local Protein Sequence Motifs Representing Common Structural Property." In: *IEEE Transactions on Nanobioscience*. Vol. 4. 3. 2005, pp. 255–265 (cit. on p. 43).

[ZM06] Arno Zinke and D. Mayer. "Applied Dynamic Programming." In: *Universitat Bonn* (2006) (cit. on p. 17).

[ZT09] Feng Zhou and Fernando Torre. "Canonical time warping for alignment of human behavior." In: *Advances in neural information processing systems*. 2009, pp. 2286–2294 (cit. on p. 83).

[ZT12] Feng Zhou and Fernando De la Torre. "Generalized time warping for multi-modal alignment of human motion." In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 1282–1289 (cit. on p. 58).

[ZTH08] Feng Zhou, F Torre, and Jessica K Hodgins. "Aligned cluster analysis for temporal segmentation of human motion." In: *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*. IEEE. 2008, pp. 1–7 (cit. on p. 58).

[ZTH13] Feng Zhou, Fernando De la Torre, and Jessica K Hodgins. "Hierarchical aligned cluster analysis for temporal clustering of human motion." In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.3 (2013), pp. 582–596 (cit. on p. 50).

Abstract — Temporal data naturally arise in various emerging applications, such as sensor networks, human mobility or internet of things. Clustering is an important task, usually applied *a priori* to pattern analysis tasks, for summarization, group and prototype extraction; it is all the more crucial for dimensionality reduction in a big data context. Clustering temporal data under time warp measures is challenging because it requires aligning multiple temporal data simultaneously. To circumvent this problem, costly k-medoids and kernel k-means algorithms are generally used. This work investigates a different approach to temporal data clustering through weighted and kernel time warp measures and a tractable and fast estimation of the representative of the clusters that captures both global and local temporal features. A wide range of public and challenging datasets, encompassing images, traces and ECG data that are non-isotropic (*i.e.*, non-spherical), not well-isolated and linearly non-separable, is used to evaluate the efficiency of the proposed temporal data clustering. The results of this comparison illustrate the benefits of the method proposed, which outperforms the baselines on all datasets. A deep analysis is conducted to study the impact of the data specifications on the effectiveness of the studied clustering methods.

Keywords: Temporal data, clustering, time warp, temporal alignment kernel, k-means, kernel k-means

Université Grenoble Alpes, Laboratoire d'Informatique de Grenoble